The Lightning Experience Guide

How to Transition to the New Salesforce

Version 4, Spring '17

Written by

Michael Alderete
Michelle Chapman-Thurber
Chris Duarte

With contributions by

Jennifer Crabtree
Cate deHeer
Brian Donnelly
Chalon Emmons
Carol Franger
Justine Heritage
Michael Hoban
Dave Jacowitz
Tammy Rahn
Samantha Ready
Erin Tidwell
Lance Santin
Jillian West
Emily Wilska
Ari Langer

This book is intended for administrators and developers who want to learn about and migrate to Salesforce's new Lightning Experience. To help you get familiar with Lightning Experience, we take you through a virtual tour of its features and functionality, and give you suggestions on how to roll it out in your organization. In addition, we've included information for developers about how Lightning Experience changes the way you develop applications, as well as detailed information about Visualforce in Lightning Experience.

The Lightning Experience Guide

CONTENTS

WELCOME TO LIGHTNING EXPERIENCE 1

Chapter 1: What Your Sales Reps Need 1

Welcome to Lightning Experience 2
What About the Previous User Experience? 3
When Lightning Experience Meets Salesforce Classic 5

Chapter 2: What Makes Lightning Experience So Special 7

Take a Tour of Lightning Experience 8
So is Lightning Experience Right for Me? 15

Chapter 3: Understanding Lightning Experience and Salesforce Classic 17

Does My Org Qualify for Lightning Experience? 18
Comparing Lightning Experience and Salesforce Classic 19
Other Features That Aren't Supported 28
What About Apex and Visualforce? 29
Best Use Cases for Lightning Experience 29
Preview Your Org in Lightning Experience 30
Evaluate Your Org with the Lightning Experience Readiness Check 30

Chapter 4: Your Lightning Experience Rollout Strategy 33

Enabling Lightning Experience 34
Recommended Features for Lightning Experience 34
Set Up Users for Lightning Experience 35
Turn Lightning Experience On! 37
Switching Between Lightning Experience and Salesforce Classic 39

WHAT'S NEW IN LIGHTNING EXPERIENCE 41

Chapter 5: Navigation and Setup in Lightning Experience 41

The Lightning Experience Navigation Bar 42
The App Launcher in Lightning Experience 43
Meet the New and Improved Setup 46

Object Manager 50
Search for Records 52
Lightning Experience Help Menu 56

Chapter 6: Opportunities, Leads, and Selling in Lightning Experience 59

Explore the Opportunity Workspace 60
Opportunity Home 63
Explore the Lead Workspace 65

Chapter 7: Explore More of Lightning Experience 67

The Home Page 68
Object Home and List Views 70
Accounts, Contacts, and Other Objects 72
User Profile 76
Tasks, Calendar, and Events 77
Chatter in Lightning Experience 81

Chapter 8: Reports and Dashboards in Lightning Experience 87

Report and Dashboard Home Pages 88
Create Beautiful Dashboards Using a Feature-Rich Editor 88
Present and Share Information in Interactive Dashboards 89
Get More Information from Interactive Charts and Filters on the Report Run Page 90
Reports and Dashboards: Compatibility Between Lightning Experience and Salesforce Classic 93

HOW YOUR EXISTING CUSTOMIZATIONS AFFECT LIGHTNING EXPERIENCE 95

Chapter 9: Your Layouts Can Customize Lightning Experience Records 95

Page Layouts in Lightning Experience 96
Compact Layouts in Lightning Experience 98

Chapter 10: Actions and Lightning Experience 101

Actions in Lightning Experience 102
How Actions Are Ordered in Lightning Experience 107

CREATE LIGHTNING APPS 109

Chapter 11: What Is a Lightning App? 109

Navigate at the Speed of Lightning 110
Find Items and Apps in the App Launcher 111
Meet the Lightning Experience App Manager 113
Install a Custom Classic App: The Warehouse Data Model 114

Chapter 12: Step into the World of Lightning Apps 117

Create a Lightning App 118
Tips for Creating Apps in Lightning Experience 119

Chapter 13: Take Your Classic App to the Next Level: Lightning 121

Upgrade the Warehouse App 122
Give the Warehouse App That Touch of Lightning 122
Look at the Warehouse App Now 124

CREATE CUSTOM PAGES FOR LIGHTNING EXPERIENCE 127

Chapter 14: Introduction to the Lightning App Builder 127

How the Lightning App Builder Works 128
The Lightning App Builder User Interface 129
Lightning Page Types 131

Chapter 15: Yes, Virginia, You Can Customize Your Lightning Experience Home Page 133

Create a Custom Home Page 135
Roll Out Your Custom Home Page to Your Lightning Experience Users 136
Look at Your Handiwork 136

Chapter 16: Those Lightning Experience Record Pages? You Can Customize Them, Too! 139

Create a Custom Lightning Record Page 140
Roll Out Your Custom Record Page to Your Lightning Experience Users 143
See What You Did There? 144

Chapter 17: The App Home Lightning Page 147

Create an App Page 148
Add Your App Page to Salesforce1 and Lightning Experience 149
Test Your App Page in Salesforce1 151
Test Your App Page in Lightning Experience 152

MIGRATE JAVASCRIPT BUTTON FUNCTIONALITY TO LIGHTNING 155

Chapter 18: JavaScript Buttons: It's Time to Move On 155

JavaScript Button Security Issues and Use Cases 156

Chapter 19: You Have Buttons? We Have Alternatives 159

Quick Actions 161
Custom URL Buttons and Links 166
Apex Triggers 167
Custom Visualforce Buttons 168

Chapter 20: Lightning Actions: Smart, Fast, and Mobile 171

Populate Fields Based on User Input and Provide Feedback Messages to Users 173
Integrate Third-Party APIs 181

ROLL OUT LIGHTNING EXPERIENCE 187

Chapter 21: Welcome to Your Lightning Experience Rollout 187

What It Takes to Go Live 188
Educate Yourself About Lightning Experience 188
Getting Hands-On with Lightning Experience 191
Identify Stakeholders and an Executive Sponsor 191
Educate Your Company About Lightning Experience 192
How to Demonstrate Lightning Experience 192
Revisit Your Processes 193
Perform a Gap Analysis 194
Present Your Findings 196
When the Show Is a Go 196

Chapter 22: All Systems Go 197

Where to Start 198
Plan Your Rollout 198
Decide Who Gets Lightning Experience 198
Unleash the Power of Super Users 199
Create a Chatter Group for the Rollout Team 201
Pick a Launch Date 202
Create a Project Schedule 203
Define Measures for Success 204
Create a Marketing and Communication Strategy 205
Create a Training Plan 206
Test Your Customizations and Iterate 207
Ready to Start 207

Chapter 23: Go Live 209

Engage with Super Users 210
Measure Results 211
Survey Your Users 211
Use Reports and Dashboards in Salesforce 213
Deliver an Executive Summary 214
The Show Must Go On 215

LIGHTNING EXPERIENCE DEVELOPMENT 217

Chapter 24: User Interface Development Considerations 217

Raising the Bar for Web App User Interfaces 218
Classic Visualforce 218
Visualforce as a JavaScript Application Container 219
Lightning Components 220
Choosing the Right Tool for the Job 221
Choosing the Right Tool for Your Organization 224
Migrating to Lightning Components 225

Chapter 25: Visualforce and Lightning Experience 227

What Works 228
What Works, But Needs Updating and Testing 229
What Doesn’t Work 229

That Look-and-Feel Thing 231

Chapter 26: Lightning Components in Lightning Experience 233

Considerations for Use 235

Chapter 27: ISVs, Packaging, and AppExchange 237

ISV Tools in Lightning Experience 238
Packaging Apps in Lightning Experience 238
AppExchange and Lightning Experience 239

Chapter 28: Understanding Changes to Other Development Tools 241

Installed Packages in Lightning Experience 242
The API and Apex in Lightning Experience 242
Authentication and Security in Lightning Experience 242
Canvas for Lightning Experience 242
Salesforce1 for Lightning Experience 243
Mobile SDK for Lightning Experience 243

VISUALFORCE & LIGHTNING EXPERIENCE 245

Chapter 29: Using Visualforce in Lightning Experience 245

Where You Can Use Visualforce in Lightning Experience 247

Chapter 30: Developing Visualforce Pages for Lightning Experience 253

Set Up Your Editor 254
Viewing Visualforce Pages During Development 254
Reviewing Visualforce Pages in Multiple Environments 258
Testing Your Visualforce Pages 259

Chapter 31: Exploring the Visualforce App Container 261

The Outer Lightning Experience Container 262
The Visualforce iframe 262
Impact of the New Container 263
Visualforce Defaults and Environment Changes in Lightning Experience 265
<apex:page> showHeader and showSidebar Attributes Are Always false 265
The sforce.one JavaScript Utility Object 265

Chapter 32: Sharing Visualforce Pages Between Classic and Lightning Experience . . . 267

Detecting and Responding to the User Experience Context in Visualforce Markup . . . 268
Detecting and Responding to the User Experience Context in JavaScript . . . 270
Determining the User Experience Context in Apex . . . 274
Querying for Lightning Experience via SOQL and API Access . . . 275

Chapter 33: Managing Navigation . . . 277

Navigation in Lightning Experience . . . 278
Classic Visualforce Navigation . . . 278
Modern Visualforce Navigation . . . 279
Navigation Gotchas, and How to Fix Them . . . 280

Chapter 34: Understanding Important Visual Design Considerations . . . 283

Affecting the Styling of Standard Components . . . 284
Styling Strategies and Recommendations . . . 285
The Salesforce Lightning Design System . . . 287

Chapter 35: Knowing Which Features to Avoid in Lightning Experience . . . 289

Lightning Experience Header and Navigation Menu Can’t Be Suppressed . . . 290
Salesforce Classic Header and Sidebar are Always Suppressed . . . 290
<apex:relatedList> and Blacklisted Related Lists . . . 290
Avoid <apex:iframe> . . . 291
No, Really, Don’t Set window.location Directly . . . 291
sforce.one Isn’t Salesforce1-Only . . . 291
Changes With Action Overrides . . . 291

INDEX . . . 293

WELCOME TO LIGHTNING EXPERIENCE

CHAPTER 1 What Your Sales Reps Need

In this chapter ...

- Welcome to Lightning Experience
- What About the Previous User Experience?
- When Lightning Experience Meets Salesforce Classic

Your sales reps have needs. They need good data in Salesforce so they can find just-in-time information about their customers and prospects. They need to work in Salesforce with as few clicks as possible, so they can sell faster and smarter. And they need the ability to do things themselves, without having to come to you, the Salesforce admin, for every request.

Your sales reps need these things so they can spend their time where it matters most: Selling your company's product and crushing their numbers.

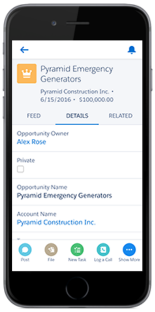
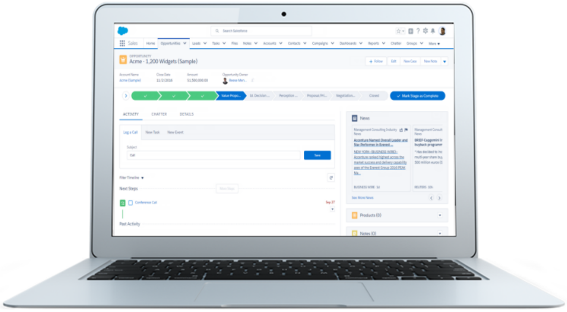
So how do you help your sales reps get what they need?

Welcome to Lightning Experience

Welcome to Lightning Experience, the new, fast, beautiful user experience from Salesforce. With a sales-centric mindset, we focused on reinventing the desktop environment to better support your sales process. The result is an intuitive, intelligent interface that helps sales reps work more naturally and close more deals faster.

Although we've started with sales, Lightning Experience won't end there. It's really just the beginning! Lightning Experience will transform Salesforce CRM and extend to service, apps, platform, and more. No matter what department you're in, we understand that the way people work is changing.

All this change is greatly influenced by the rise of mobile. Sales reps at your company are already using mobile to research prospective customers, get directions to client meetings, connect socially with customers, and more. We get that. That's why Lightning Experience takes the cool stuff from the mobile experience and brings it to the desktop.

What sales reps love about mobile…	…becomes what sales reps love about desktop
 Pyramid Emergency Generators Pyramid Construction Inc. • 6/15/2016 • $100,000.00 FEED DETAILS RELATED Opportunity Owner Alex Rose Private □ Opportunity Name Pyramid Emergency Generators Account Name Pyramid Construction Inc. Post File New Task Log a Call Show More	 Search Salesforce Sales Home Opportunities Leads Tasks Files Notes Accounts Contacts Campaigns Dashboards Reports Chatter Groups More OPPORTUNITY Acme - 1,200 Widgets (Sample) + Follow Edit New Case New Note Account Name Acme (Sample) Close Date 11/1/2016 Amount $1,500,000.00 Opportunity Owner Value Propo… Id. Decision… Perception … Proposal/Pri… Negotiation… Closed ✓ Mark Stage as Complete ACTIVITY CHATTER DETAILS Log a Call New Task New Event Subject Call Save Filter Timeline Next Steps Conference Call Sep 27 Past Activity News Management Consulting Industry News Accenture Named Overall Leader and Star Performer in Everest… NEW YORK--(BUSINESS WIRE)-- Accenture ranked highest across the market success and delivery capability axes of the Everest Group 2016 PEAK Ma… BUSINESS WIRE 1d See More News Products (0) Notes (0)

When we're talking about Lightning Experience, we're talking about pages in Salesforce optimized for sales use. We're talking about new features that help your sales reps focus on the right deals and the right activities, every time they log in. We're talking about flexible, interactive tools that sales reps can use to visualize data and work deals in flight.

But before we go any further, let's talk about how Lightning Experience got its name, and why we built it in the first place.

Why We Built Lightning Experience

Let's start with the name, and while we're at it, the correct spelling of the name.

It's Lightning!	Not Lightening	Or Lighting
A modern, productivity-boosting user experience designed to help sales reps close deals lightning fast (see what I did there?)	The act of becoming lighter, like when you put down that swag-stuffed backpack you're carrying at Dreamforce	The method by which something is illuminated, like the soft glow of a lamp, by the light of which you read the release notes

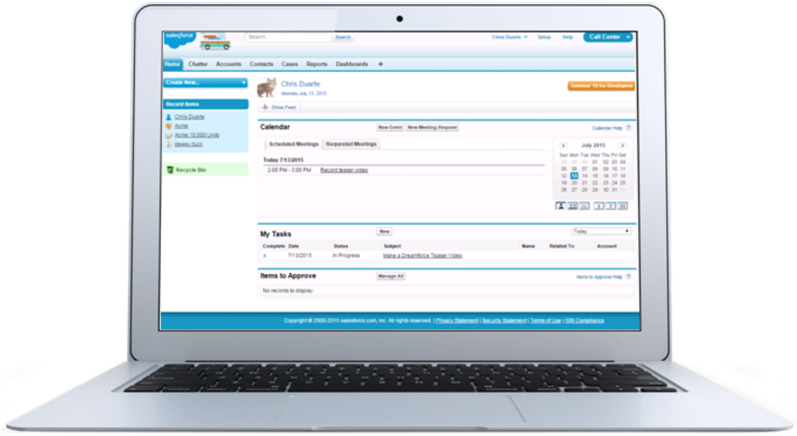
So why the name "Lightning?" Well, think for a moment about actual lightning, the kind you see during a storm. Think about how fast it strikes; if you blink, you might miss it. Think about how beautiful it is; lightning can be stunning to behold. Finally, think about how unique each lightning bolt is; no two are the same.

That's a lot like Lightning Experience. It's fast, it's beautiful, and it's unique to each sales rep. It's a simpler user experience, designed to help sales reps sell faster, with personalized alerts and an interactive assistant to help each sales rep focus on what's important. All of this drives improved productivity for your sales team.

Ultimately, we built Lightning Experience because of you, our customer. Lightning Experience is the result of everything we've learned from you over the past years and releases. Lightning Experience is the killer sales app you can deliver to your sales reps.

What About the Previous User Experience?

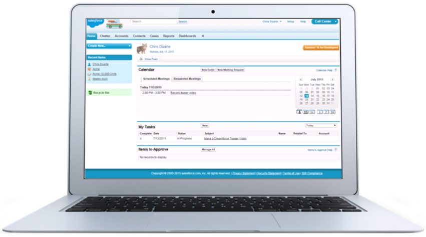
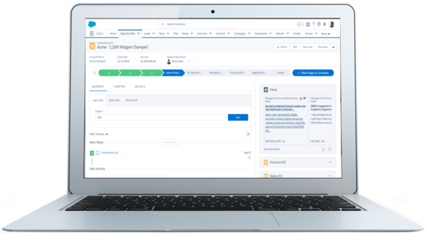
It's still here, and for the purposes of clarity, let's give this user experience a name: Salesforce Classic.



Salesforce Classic refers to the Salesforce user experience immediately predating Lightning Experience. It's the one with the Force.com App Menu and the user name, Setup, and Help links in the top right. If you haven't customized your logo, you also see the seasonal Salesforce logo in the upper left corner when you're looking at Salesforce Classic.

So that's Salesforce Classic, and again, it's still here. It has to still be here, because not all features are available in Lightning Experience yet, and besides, you might not be ready to move off Salesforce Classic just now. Depending on your company culture, change can take time, and we get that, too. We'll talk more about change management and migration in another module, since that's an important topic all on its own.

Back to Salesforce Classic and Lightning Experience. The first thing you need to know is how to tell when you're in one or the other.

The Force.com App Menu in the top right means you're in Salesforce Classic	**Icons in the top right mean you're in Lightning Experience**
	

It's important that you get comfortable with identifying which one you're in. If you decide to enable Lightning Experience, you need to prepare for helping users with questions. As a Salesforce admin who has enabled Lightning Experience, one of the first things you'll need to identify on a troubleshooting call with a user is, are you in Lightning Experience or in Salesforce Classic?

We mentioned earlier that some features are not yet available in Lightning Experience, and right about now you're probably wondering exactly how that works. Let's talk about that next.

When Lightning Experience Meets Salesforce Classic

Features that aren't supported in Lightning Experience don't appear in Lightning Experience. But don't fear! You can still get to those features using the Switcher, which we'll talk about later.

If any of your users' primary Salesforce use cases require currently unsupported features, those users probably aren't good candidates for Lightning Experience, and you might want to exclude them from any Lightning Experience pilots for now. The more complex your sales process, the more likely it is that your users may need to stick with Salesforce Classic for a while.

Note: For a detailed list of supported and unsupported features, see the Salesforce Help.

So what do you do if you have sales reps who would benefit from Lightning Experience, but service reps and other employees who primarily use features not yet available in the new interface? Depending on your edition, you can deploy Lightning Experience to a set of users, using profiles or permission sets. We'll cover this in detail later, but for now, let's talk about the benefits you get once you deploy Lightning Experience.

CHAPTER 2 What Makes Lightning Experience So Special

In this chapter ...

- Take a Tour of Lightning Experience
- So is Lightning Experience Right for Me?

With all its new features and redesigned pages, there are many key benefits to using Lightning Experience. We won’t cover everything here, but for each area of the product, we’ll highlight some of the neatest things.

To kick things off, let’s talk about some of the highlights:

- Efficient navigation and the ability to switch between custom-branded apps
- Quick access to productivity tools like Notes and Recent Items in the utility bar
- New record layouts that focus on what you can do instead of what you can view
- Turbocharged list views that let you easily filter and visualize your data
- Beautiful dashboards with components that span both columns and rows
- Sleek report views that you can filter quickly to see the data that's most important to you

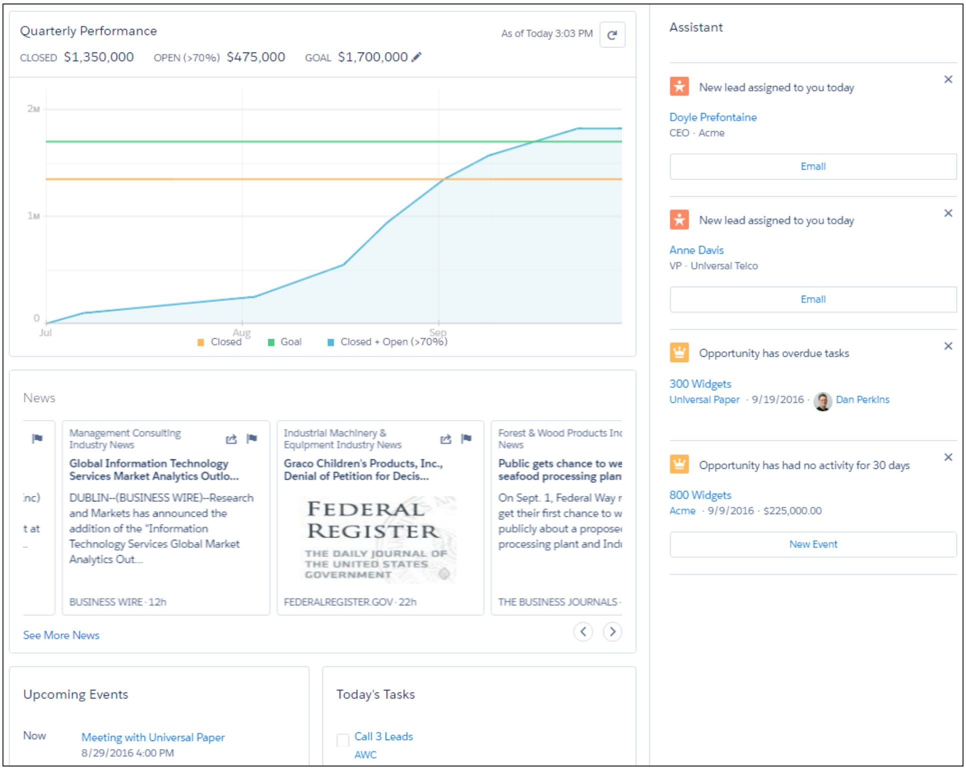
Take a Tour of Lightning Experience

Why is Lightning Experience so special? We'll show you!

Home

We've reimagined the way you start your day with a brand new, intelligent page. Now your sales reps can monitor their performance to goal and get insights on their key accounts. Plus, we've added an Assistant, which is your users' action list of things to do. You can also use the Lightning App Builder to create custom Home pages that appear for different profiles.

- Start your day fast with a customizable, intelligent page.
- Use the Performance Chart to monitor how close you are to crushing your numbers.
- See relevant, timely news articles about customers, partners, and competitors with News.
- See upcoming meetings and tasks due today.
- Use the Assistant to identify key issues to work on today.
- Focus your selling activities on your Top Deals.



Quarterly Performance

As of Today 3:03 PM

CLOSED $1,350,000 OPEN (>70%) $475,000 GOAL $1,700,000

News

Management Consulting Industry News

Global Information Technology Services Market Analytics Outlo...

DUBLIN--(BUSINESS WIRE)--Research and Markets has announced the addition of the "Information Technology Services Global Market Analytics Out...

BUSINESS WIRE · 12h

Industrial Machinery & Equipment Industry News

Graco Children's Products, Inc., Denial of Petition for Decis...

FEDERAL REGISTER THE DAILY JOURNAL OF THE UNITED STATES GOVERNMENT

FEDERALREGISTER.GOV · 22h

Forest & Wood Products Inc News

Public gets chance to we seafood processing plan

On Sept. 1, Federal Way r get their first chance to w publicly about a propose processing plant and Indu

THE BUSINESS JOURNALS ·

See More News

Upcoming Events

Now Meeting with Universal Paper 8/29/2016 4:00 PM

Today's Tasks

□ Call 3 Leads AWC

Assistant

New lead assigned to you today

Doyle Prefontaine CEO · Acme

Email

New lead assigned to you today

Anne Davis VP · Universal Telco

Email

Opportunity has overdue tasks

300 Widgets Universal Paper · 9/19/2016 · Dan Perkins

Opportunity has had no activity for 30 days

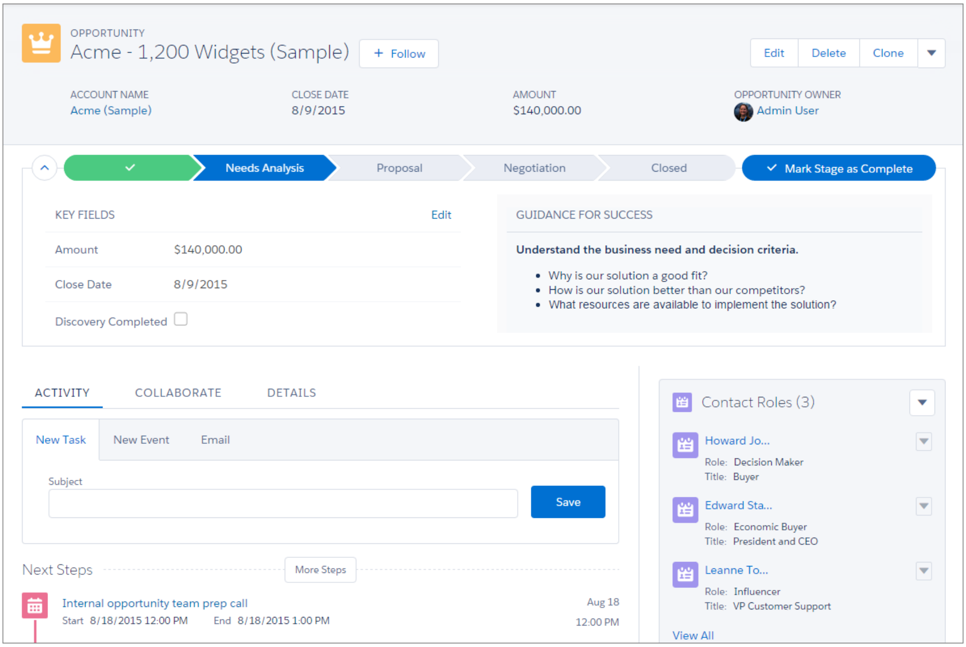
800 Widgets Acme · 9/9/2016 · $225,000.00

New Event

Opportunity Workspace

We've taken your sales process and put it into an action-optimized workspace, designed to help your sales reps work their deals faster and smarter. You can customize coaching scripts for each step in the sales process, create records quickly with fewer clicks, and ultimately close deals faster.

- Showcase key record details in the new highlights panel at the top of the page.
- Use the handy composer to quickly log calls, create tasks, send emails, and more.
- Get key coaching details with a customizable sales path to support your sales process.
- See a wealth of related information on hover using quick view, without ever leaving the opportunity page.
- Add related records—like contacts—in context with minimal clicks.
- View relevant, timely news about the account the opportunity's associated with.



OPPORTUNITY
Acme - 1,200 Widgets (Sample)

ACCOUNT NAME: Acme (Sample)
CLOSE DATE: 8/9/2015
AMOUNT: $140,000.00
OPPORTUNITY OWNER: Admin User

KEY FIELDS

Amount	$140,000.00
Close Date	8/9/2015
Discovery Completed	□

GUIDANCE FOR SUCCESS

Understand the business need and decision criteria.

- Why is our solution a good fit?
- How is our solution better than our competitors?
- What resources are available to implement the solution?

Subject

Next Steps

Internal opportunity team prep call
Start 8/18/2015 12:00 PM End 8/18/2015 1:00 PM
Aug 18
12:00 PM

Contact Roles (3)

Howard Jo...
Role: Decision Maker
Title: Buyer

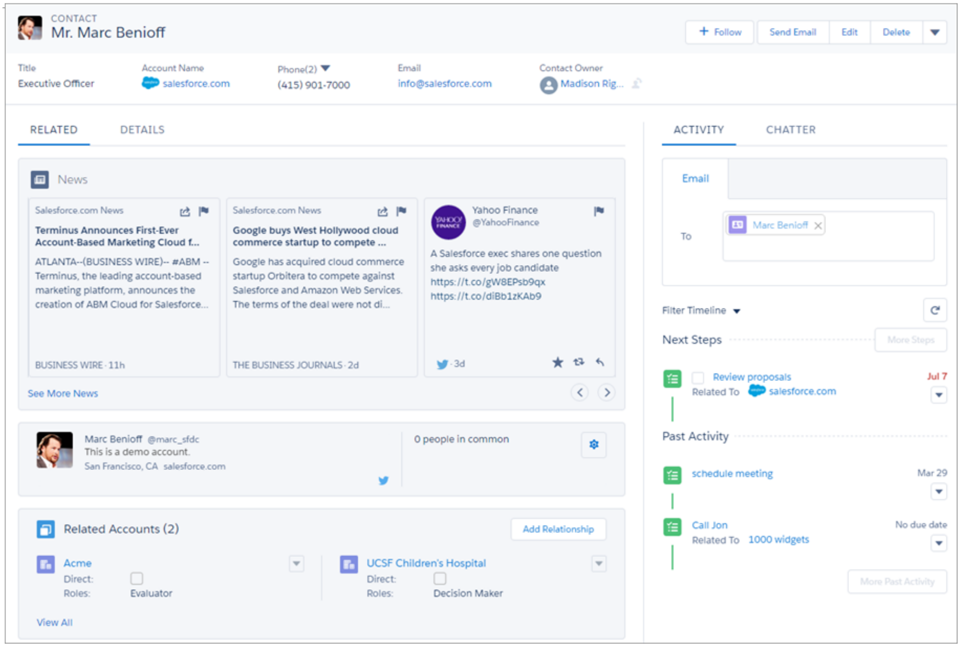
Edward Sta...
Role: Economic Buyer
Title: President and CEO

Leanne To...
Role: Influencer
Title: VP Customer Support

Accounts and Contacts

Like with opportunities (and leads), we've optimized the layout for accounts and contacts, organizing the content by their primary use case: reference. Now your sales reps can find information and gather information at a glance.

- Locate important data efficiently with the redesigned Lightning Experience page layout.
- Get the latest news for your customers with News and integrated Twitter.
- Work smarter and keep your data clean with field-level duplicate matching.
- Review past and upcoming activities at a glance.



CONTACT
Mr. Marc Benioff

Title: Executive Officer
Account Name: salesforce.com
Phone(2): (415) 901-7000
Email: info@salesforce.com
Contact Owner: Madison Rig...

News

Salesforce.com News
Terminus Announces First-Ever Account-Based Marketing Cloud f...
ATLANTA--(BUSINESS WIRE)-- #ABM -- Terminus, the leading account-based marketing platform, announces the creation of ABM Cloud for Salesforce...
BUSINESS WIRE · 11h

Salesforce.com News
Google buys West Hollywood cloud commerce startup to compete ...
Google has acquired cloud commerce startup Orbitera to compete against Salesforce and Amazon Web Services. The terms of the deal were not di...
THE BUSINESS JOURNALS · 2d

Yahoo Finance @YahooFinance
A Salesforce exec shares one question she asks every job candidate https://t.co/gW8EPsb9qx https://t.co/diBb1zKAb9
· 3d

Marc Benioff @marc_sfdc
This is a demo account.
San Francisco, CA salesforce.com

0 people in common

Related Accounts (2)

Acme
Direct: □
Roles: Evaluator

UCSF Children's Hospital
Direct: □
Roles: Decision Maker

Email
To: Marc Benioff

Next Steps

□ Review proposals — Jul 7
Related To salesforce.com

Past Activity

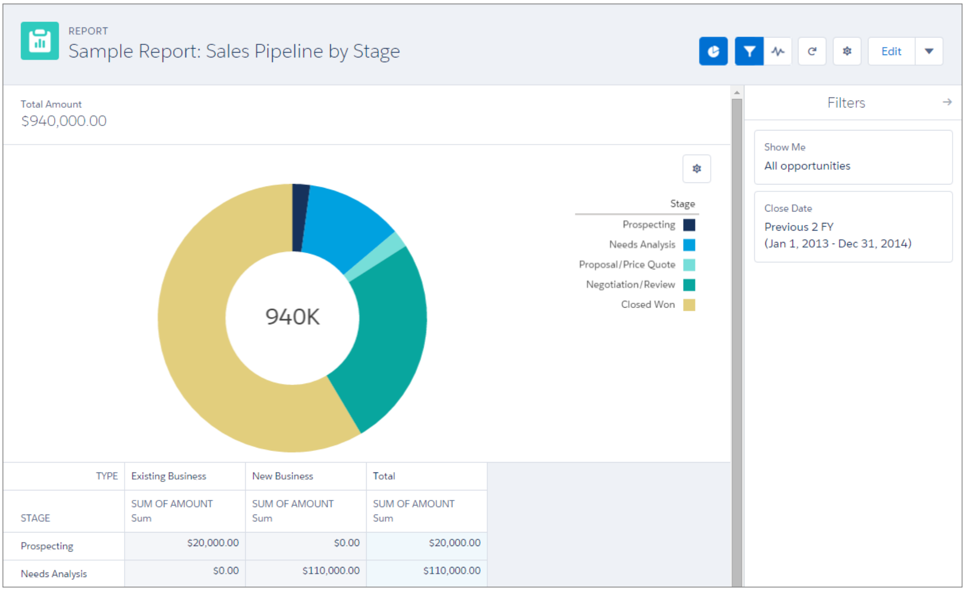
schedule meeting — Mar 29

Call Jon — No due date
Related To 1000 widgets

Reports and Dashboards

Sales reps will love the ability to create their own filters on reports. You will appreciate the updated dashboard editor, which features components that span both columns and rows.

- Create filters while viewing a report.
- Make visually awesome dashboards with more than three columns.
- Easy migration from Salesforce Classic to Lightning Experience, with reports and dashboards automatically viewable and inheriting all permissions and sharing settings defined in Salesforce Classic.



REPORT

Sample Report: Sales Pipeline by Stage

Total Amount
$940,000.00

Filters

Show Me
All opportunities

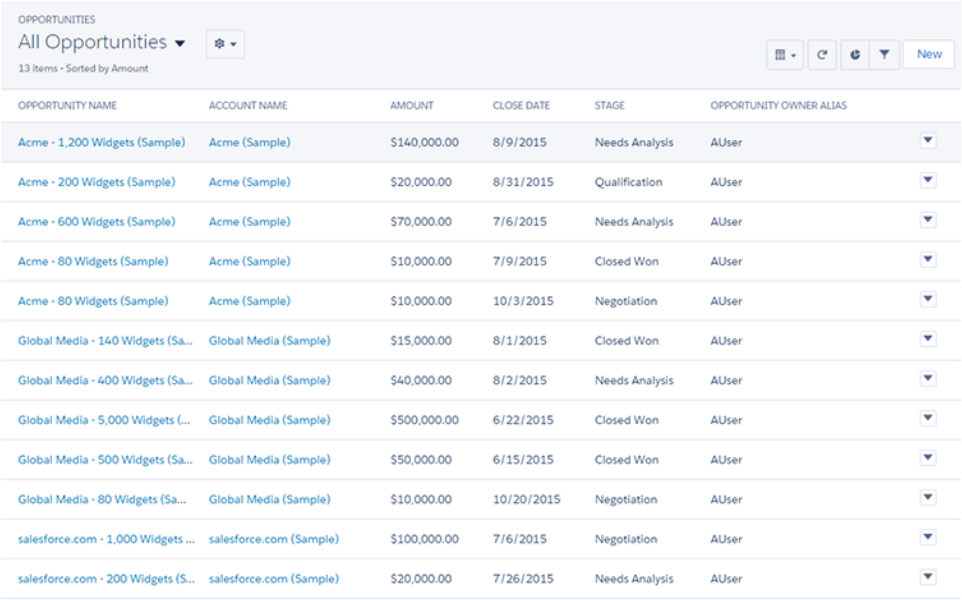
Close Date
Previous 2 FY
(Jan 1, 2013 - Dec 31, 2014)

TYPE	Existing Business	New Business	Total
STAGE	SUM OF AMOUNT Sum	SUM OF AMOUNT Sum	SUM OF AMOUNT Sum
Prospecting	$20,000.00	$0.00	$20,000.00
Needs Analysis	$0.00	$110,000.00	$110,000.00

List Views

Now your sales reps can visualize any list view graphically with a handy chart, or easily apply filters to narrow the results.

- Visualize your data in seconds with list view charts, and quickly create filters to slice your data how you want.
- Create list views by using Lightning Experience's intuitive filters panel.
- Use type-ahead search to find a favorite list view fast.
- Automatically open your list views created in Salesforce Classic from Lightning Experience.
- Use inline editing to make quick changes to records in a list view.



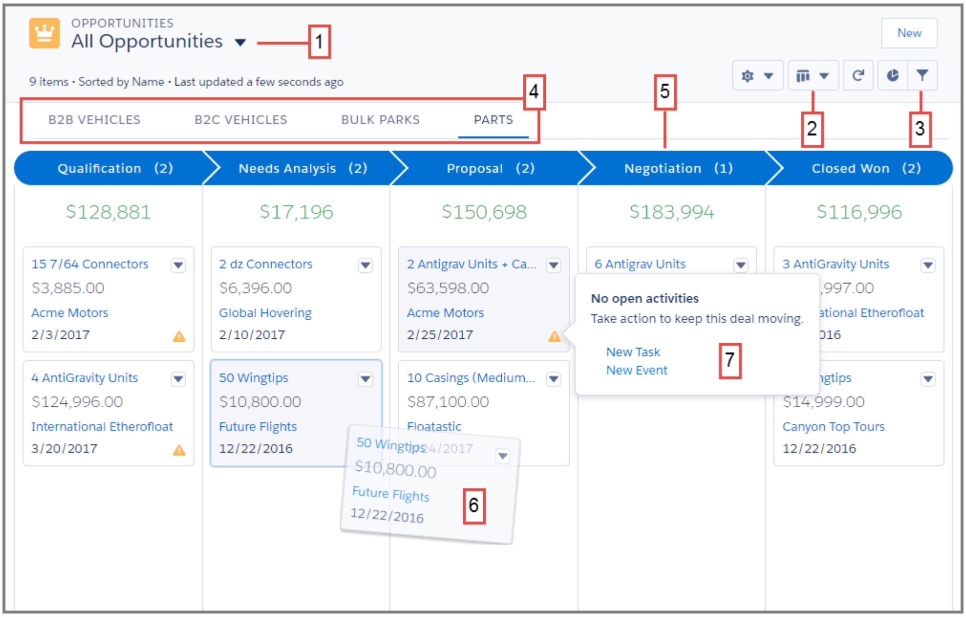
13 items • Sorted by Amount

OPPORTUNITY NAME	ACCOUNT NAME	AMOUNT	CLOSE DATE	STAGE	OPPORTUNITY OWNER ALIAS
Acme - 1,200 Widgets (Sample)	Acme (Sample)	$140,000.00	8/9/2015	Needs Analysis	AUser
Acme - 200 Widgets (Sample)	Acme (Sample)	$20,000.00	8/31/2015	Qualification	AUser
Acme - 600 Widgets (Sample)	Acme (Sample)	$70,000.00	7/6/2015	Needs Analysis	AUser
Acme - 80 Widgets (Sample)	Acme (Sample)	$10,000.00	7/9/2015	Closed Won	AUser
Acme - 80 Widgets (Sample)	Acme (Sample)	$10,000.00	10/3/2015	Negotiation	AUser
Global Media - 140 Widgets (Sa...	Global Media (Sample)	$15,000.00	8/1/2015	Closed Won	AUser
Global Media - 400 Widgets (Sa...	Global Media (Sample)	$40,000.00	8/2/2015	Needs Analysis	AUser
Global Media - 5,000 Widgets (...	Global Media (Sample)	$500,000.00	6/22/2015	Closed Won	AUser
Global Media - 500 Widgets (Sa...	Global Media (Sample)	$50,000.00	6/15/2015	Closed Won	AUser
Global Media - 80 Widgets (Sa...	Global Media (Sample)	$10,000.00	10/20/2015	Negotiation	AUser
salesforce.com - 1,000 Widgets ...	salesforce.com (Sample)	$100,000.00	7/6/2015	Negotiation	AUser
salesforce.com - 200 Widgets (S...	salesforce.com (Sample)	$20,000.00	7/26/2015	Needs Analysis	AUser

Kanban View

The Kanban view organizes a set of records into columns so track your work at a glance. To update a record's status, drag it into a different column. You can configure the board by selecting what fields columns and summaries are based on. And, get personalized alerts on key opportunities in flight.

- Visualize your work at each stage or status
- Move records between columns using drag and drop functionality
- Configure columns and summary fields on the fly
- Edit or delete records to keep them up to date
- Quickly create filters to slice your data how you want
- For opportunities, get alerts to notify you when action is needed on a key deal



1. The records in the Kanban view are based on the selected list view.
2. Easily toggle between the list view grid view and the Kanban view.
3. Filter your records to view a particular subset of your records.
4. Select which record type to view.
5. Columns are created based on the grouping field.
6. Quickly move a record to a different column by dragging the card.
7. For opportunities, alerts tell how to keep a deal on track, for example, create a task or event.

Other Highlights

- Make your users more efficient and allow them to switch between apps that you can brand and customize. Create records and access recent records and lists for certain items directly from the navigation bar.
- See open tasks, planned meetings, and accomplishments in the activity timeline on each opportunity, lead, account, and contact.
- Find your tasks on a new page for tasks, including the new master-detail view, which lets you see a single task and your entire list of tasks side by side.

- Use phone features without ever leaving Salesforce. You can make and receive calls, add call notes, and log call information with ease.
- Send email through Gmail or Office 365 accounts with your Salesforce email. See the emails you've sent in your Gmail or Office 365 Sent Items folder for seamless integration.
- Take better notes with auto-save, rich text capabilities, inline images, and versioning. You can even relate a note to multiple records and share notes with teammates or Chatter groups.
- Enjoy a richer file preview experience that doesn't require Adobe Flash Player.
- Find records faster with improved search, which includes recent records and top results.

So is Lightning Experience Right for Me?

At this point, you're probably starting to think about whether or not Lightning Experience is a good fit for your company. This is good, and part of the point of this content is to help you make that choice.

Ultimately, your decision comes down to this: does the rad stuff you get with Lightning Experience outweigh what you can't do without Salesforce Classic?

So, let's help you decide. Next, we'll take you through the key differences between the two interfaces and help you figure out how to make that decision.

CHAPTER 3 Understanding Lightning Experience and Salesforce Classic

In this chapter ...

- Does My Org Qualify for Lightning Experience?
- Comparing Lightning Experience and Salesforce Classic
- Other Features That Aren't Supported
- What About Apex and Visualforce?
- Best Use Cases for Lightning Experience
- Preview Your Org in Lightning Experience
- Evaluate Your Org with the Lightning Experience Readiness Check

Lightning Experience is a modern user interface for your sales reps to help them sell faster and smarter. Lightning Experience includes many new features and entirely redesigned pages, but not every Salesforce feature is supported in Lightning Experience. So the Salesforce experience you've known to date—called Salesforce Classic—is still available for you, and the users you enable for Lightning Experience can switch between the two at will.

As your company's trusted advisor for Salesforce, you need to help your company decide when to enable Lightning Experience. So, we've got to start getting you educated on Lightning Experience features versus Salesforce Classic.

Does My Org Qualify for Lightning Experience?

Let's look at the supported editions and user licenses to see if you can enable and use Lightning Experience.

Table 1: Salesforce Editions

Supported Editions	Unsupported Editions
• Group Edition • Professional Edition • Enterprise Edition • Performance Edition • Unlimited Edition • Developer Edition	• Personal Edition • Contact Manager Edition • Database.com Edition • Chatter.com Edition

Table 2: Salesforce User Licenses

Supported Licenses	Unsupported Licenses
• Salesforce • All Salesforce Platform and Force.com (excluding Force.com Free) • Identity User • Company (Employee) Community • Chatter External • Chatter Plus • Chatter Free	• Customer Community, Customer Community Plus, Partner Community • All Portal licenses • Force.com Free • Work.com • Database.com • Content Only • Ideas Only • Knowledge Only • Site.com Only

Comparing Lightning Experience and Salesforce Classic

Understanding what you get—and don't get—in the new interface is a big part of the decision to switch to Lightning Experience. Here are some highlights of how Lightning Experience compares to what you're used to in Salesforce Classic.

Note: This isn't a comprehensive list of supported and unsupported features. For a more detailed list, see the Salesforce Help.

Salesforce Data

Feature	Lightning Experience	Salesforce Classic
Sales objects: Accounts, Campaigns, Contacts, Leads, Opportunities, Person Accounts, Price Books, Products	✓	✓
Calendar, Events, Tasks	✓	✓
Service objects: Assets, basic Cases, Work Orders	✓	✓
Custom objects	✓	✓
Chatter feeds, groups, and people	✓	✓
Salesforce Files	✓	✓
Other Sales and Service objects		✓

Home

An intelligent page filled with insights, a slick performance chart, and a helpful assistant feature, highlighting what's most important each day.

Feature	Lightning Experience	Salesforce Classic
Performance chart	✓	
Assistant	✓	

Feature	Lightning Experience	Salesforce Classic
News	✓	
Top deals	✓	
Tasks	✓	✓
Events*	✓	✓
Calendar*	✓	✓
Feed and publisher		✓
Items to approve	✓	✓
Customizable dashboards	✓	✓
Customizable home page components	✓	✓

* In Lightning Experience, Home displays only events remaining on a user's calendar today. Users can access all their events from the Calendar tab in the navigation bar.

Accounts and Contacts

Sales reps get a layout designed to help them find the information they need, fast.

Feature	Lightning Experience	Salesforce Classic
News	✓	
Activity timeline	✓	
Enhanced Notes	✓	
“Reference” page layout	✓	
Related lists	✓	✓
Integrated email and templates	✓	✓
Twitter highlights	✓	✓

Feature	Lightning Experience	Salesforce Classic
Person accounts	✓ (Beta Support)	✓
Account teams	✓	✓
Account hierarchy	✓	✓
Contact hierarchy	✓	
Contacts to Multiple Accounts	✓	✓
Find and merge duplicate accounts	✓	✓
Find and merge duplicate contacts	✓	✓
Matching and duplicate rules for business and person accounts	✓	✓
Matching and duplicate rules for contacts	✓	✓

Campaigns (Beta)

Feature	Lightning Experience	Salesforce Classic
Campaign member status charts	✓	
Import campaign members	✓	✓
Add individual campaign members	✓	✓
Campaign member status updates via import	✓	✓
Clone campaign members	✓	✓
Add and manage campaign members from a list view	✓	✓

Opportunities and Leads

Sales reps in Lightning Experience find a turbo-charged workspace for managing opportunities and leads. View related information in the Quick View. Create tasks and events, log calls, and send emails, and track all those activities in the Activity Timeline.

Opportunities

Feature	Lightning Experience	Salesforce Classic
Workspace page layout	✓	
Activity timeline	✓	
Path	✓	
News	✓	
Enhanced Notes	✓	
Visual view of opportunities (Kanban)	✓	
Integrated email and templates	✓	✓
Opportunity teams	✓	✓
Opportunity splits		✓
Similar opportunities		✓
Big deal alerts		✓

Leads

Feature	Lightning Experience	Salesforce Classic
Workspace page layout	✓	
Activity timeline	✓	
Path	✓	
News	✓	
Enhanced Notes	✓	

Feature	Lightning Experience	Salesforce Classic
Change lead owner to a queue	✓	✓
Integrated email and templates	✓	✓
Lead conversion customization via the API	✓	✓
Find and merge duplicate leads	✓	✓
Matching and duplicate rules for leads	✓	✓

Opportunity Kanban

Sales reps can use the Opportunity Kanban, a visualization tool for opportunities, to review deals organized by each stage in the pipeline. With drag-and-drop functionality, sales reps can move deals from one stage to another, and get personalized alerts on key deals in flight.

Feature	Lightning Experience	Salesforce Classic
Charts	✓	
Drag and drop	✓	
Intelligent Alerts	✓	
Type-ahead list view search	✓	
Sharing settings*	✓	

* Only list views marked as “Visible only to me” or “Visible to all users” are supported. Sharing with certain groups of users isn’t supported.

Data.com

Feature	Lightning Experience	Salesforce Classic
Prospect for accounts (Only available with a Data.com Prospector license)	✓	✓

Feature	Lightning Experience	Salesforce Classic
Prospect for contacts (Only available with a Data.com Prospector license) Note: In Lightning Experience, you can search for and add contact data only if the contact is related to an existing account.	✓	✓
Prospect for companies in the same company hierarchy (Only available with a Data.com Prospector license)	✓	✓
Prospecting Insights (Only available with a Data.com Prospector license)	✓	✓
View fields updated by Data.com Clean	✓	✓
"Stare and Compare" using Data Integration	✓	
Manually update a record using Data Integration	✓	✓
Clean button		✓
Prospect for companies, contacts, and leads using the **Data.com** tab		✓
View data integration rule settings and status (As of the Spring '17 release, Clean rules are known as data integration rules.)	✓	✓
Add Data.com contacts as leads (Only available with a Data.com Prospector license)		✓
See match rates for data integration rules (As of the Spring '17 release, Clean rules are known as data integration rules.)	✓	✓
Set up Clean settings, jobs, and preferences		✓
View Clean metrics and analytics		✓

Feature	Lightning Experience	Salesforce Classic
Export Data.com records to comma-separated values (CSV) files. (Only available with a Data.com Prospector license)		✓

Reports

Sales reps who use reports find an improved user interface, including the ability to easily create filters and add enhanced charts. And Salesforce admins find features on the dashboard editor, including flexible layouts and spanning columns. However, some report features aren't yet available in Lightning Experience, including feeds and scheduled reports.

Feature	Lightning Experience	Salesforce Classic
Enhanced report charts	✓	
Hide totals and subgroups from report view	✓	
Interactive filters when viewing reports	✓	
Report Builder	✓	✓
Bucket fields	✓	✓
Custom summary formulas	✓	✓
Matrix, Summary, Tabular report formats*	✓	✓
Role hierarchy filters	✓	✓
Create report folders		✓
Joined reports		✓
Pie charts		✓
Schedule report refreshes**	✓	✓
Follow reports		✓
Report notifications		✓
Export reports	✓	✓

* You can't set up folder sharing in Lightning Experience; however, folders that were created in Salesforce Classic inherit all assigned permissions when viewed in Lightning Experience. **Available in Lightning Experience by Subscribing to a report.

Dashboards

Feature	Lightning Experience	Salesforce Classic
Display more than 3 columns	✓	
Dashboard Builder	✓	✓
View filtered dashboards	✓	✓
View dynamic dashboards	✓	✓
Schedule dashboard refreshes*		✓
Post dashboard components to feeds	✓	✓
Follow dashboards	✓	✓
Visualforce components		✓

* Users can't schedule dashboards in Lightning Experience, but dashboards scheduled in Salesforce Classic run as expected in Lightning Experience.

List Views

List Views have gotten an upgrade in Lightning Experience, including some super new stuff like the ability to create a quick chart from a list view or add filters with an improved, easy-to-use interface.

Feature	Lightning Experience	Salesforce Classic
Charts	✓	
Create filters on the fly	✓	
Search for a list view on the fly	✓	
View records visually (Kanban)	✓	

Feature	Lightning Experience	Salesforce Classic
Create and edit lists	✓	✓
Sortable columns	✓	✓
Resizable columns	✓	✓
Sharing Settings*	✓	✓
Filter logic	✓	✓
Inline editing	✓	✓

* In Lightning Experience, sharing list views marked as "Visible only to me" or "Visible to all users" is supported. Sharing with certain groups of users isn't supported.

Other Features and Products

Feature	Lightning Experience	Salesforce Classic
Einstein	✓	
Custom branding of Salesforce apps	✓	
Activities-related composer windows	✓	
Create and edit records	✓	✓
Inline editing record detail fields	✓	✓
Inline editing in lists	✓	✓
Customizable Forecasting		✓
Collaborative Forecasting	✓	✓
Territory Management		✓
Salesforce Communities*	✓	✓
Partner Portals		✓
Service Cloud**	✓	✓

Feature	Lightning Experience	Salesforce Classic
Work.com		✓

* Switch to Salesforce Classic for the global header, creating or disabling community users, logging in as a community user, and using delegated administration.

Other Features That Aren't Supported

There are some other features that don't work in Lightning Experience. We want to be transparent with you about them, so you can make an informed decision about when to enable Lightning Experience.

Note: This isn't a comprehensive list of supported and unsupported features. For a more detailed list, see the Salesforce Help.

Custom JavaScript Buttons

Custom buttons that use a JavaScript content source aren't supported in Lightning Experience. Check out Lightning Alternatives to JavaScript Buttons in Trailhead for a great list of Lightning-friendly solutions for your JavaScript button use cases.

AppExchange Apps with Custom Objects, Visualforce Pages, and More

Many apps from the AppExchange feature customizations, including custom objects, custom buttons, Visualforce pages, and more. Salesforce partners are testing their apps in Lightning Experience and applying for *Lightning Ready* certification. Apps that are Lightning Ready work in Lightning Experience and offer a more consistent experience with other Lightning Pages. If an app is supported in Lightning Experience, a Lightning Ready sash appears on its AppExchange listing. If an app isn't supported in Lightning Experience, use it in Salesforce Classic instead.

Note: Your org can use apps without Lightning Ready certification in Lightning Experience, but some features might not be available or work as expected. These apps can also appear visually inconsistent with Lightning Experience because they use Salesforce Classic styling. For the best experience, use Salesforce Classic for apps that aren't certified as Lightning Ready.

Customizing Tabs

Users working in Salesforce Classic may be used to customizing the tabs they see across the top of their screen in Salesforce. Similar to Salesforce Classic, the navigation bar in Lightning Experience gives your users access to sets of objects and tabs. Think of the navigation bar as a container, where the items within it change to represent the app you're using. The apps you made for Salesforce Classic work in Lightning Experience without any modifications. When you create a new app or upgrade a Classic app in Lightning Experience, you can even brand the app with a custom color and logo.

Tip: Although your users can't customize tabs, they can personalize their Lightning Experience navigation with favorites. Like bookmarks in a web browser, favorites allow quick access to important records, lists, groups, dashboards, and other frequently used pages in Salesforce.

There are some items that can't go in custom navigation menus, including connected apps and some Salesforce objects that aren't supported in Lightning Experience. But you and your users can still get to these items via the App Launcher. We'll look at custom navigation and the App Launcher later.

What About Apex and Visualforce?

When we talk about customizing Salesforce "programmatically," we're talking about using code to add new features. Most of the time, like when you're using the API or writing Apex triggers, there isn't an actual user interface to the feature. Guess what? All of that stuff *just works* whether you're in Salesforce Classic or Lightning Experience. Woohoo!

Other features do have a user interface. In that case, we're mostly talking about Visualforce. Your existing Visualforce pages work in Lightning Experience, many without needing any revisions. Because things have moved around in the new interface, you'll want to complete some chores to make sure your Visualforce pages work the way you expect.

There are some other considerations for Visualforce pages and apps, and we recommend that you review those. We created an entire Trailhead module just for Visualforce developers, and an entire *trail* for developing Lightning Experience apps. If you're using Visualforce or other programmatic customizations in your organization, check out those trails as part of your Lightning Experience assessment process.

Best Use Cases for Lightning Experience

If you've gone through the list of supported features, and you've done an assessment of what's in your organization, and you've weighed all the rad stuff you'll get against what you can't live without, then you're probably starting to decide if you're ready to enable Lightning Experience. But if you're still not sure, here's one last comparison chart.

Lightning Experience might be right for some or all of your org if:	Salesforce Classic might be right for you if:
• Your sales team does business-to-business sales using accounts, contacts, leads, opportunities, custom objects, and the other sales features supported in the new user interface. • You want to pilot the new user interface with a group of sales reps. • You're looking to reboot your Salesforce implementation. This is a great opportunity to introduce new features because you're doing change management anyway.	• Your sales team makes regular use of features that aren't yet available in Lightning Experience, such as forecasting, or territory management. • You want a single experience for your sales and service teams.

Preview Your Org in Lightning Experience

It's all well and good to talk about use cases and supported and unsupported features. But wouldn't it be great if you could see your Salesforce org in Lightning Experience without having to enable the new interface? Even if you don't have a sandbox org? Well, the great news is you can, using the Lightning Experience Migration Assistant. The Migration Assistant's Preview feature lets you explore your production org in Lightning Experience, so you can see exactly how your real data and your current customizations work in the new interface.

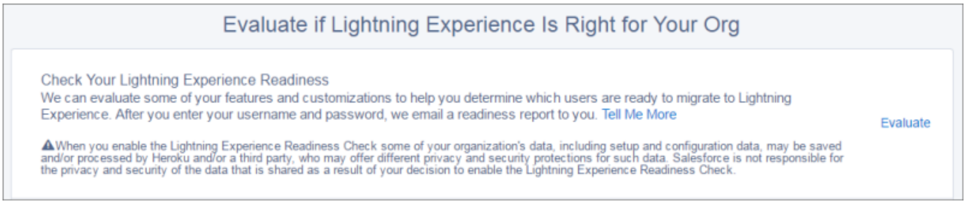
The Preview feature isn't a locked-down, read-only tour. You can change data, settings, permissions—you name it. But remember that you're working with your live org. If you do make changes, they're for real and visible to your users back in Salesforce Classic.

To get started, from Setup, enter *`Lightning`* in the `Quick Find` box, then select **Lightning Experience**. Click **Preview** to dive in.

Evaluate Your Org with the Lightning Experience Readiness Check

Run the Lightning Experience Readiness Check to evaluate some of the Sales Cloud features and customizations you're using in Salesforce Classic. That way, you can learn which are ready for Lightning Experience and which need your attention before you migrate your users.

After you kick off the evaluation, you get a report with an impact assessment on the features and customizations we evaluated. We also give you our recommendations on taking next steps. The readiness check evaluates several of your Sales Cloud features, including unsupported tabs, objects, custom buttons, and links. You can kick off the Lightning Experience Readiness Check from the Migration Assistant in Setup. Then, we email you your readiness report.



Evaluate if Lightning Experience Is Right for Your Org

Check Your Lightning Experience Readiness

We can evaluate some of your features and customizations to help you determine which users are ready to migrate to Lightning Experience. After you enter your username and password, we email a readiness report to you. Tell Me More

⚠When you enable the Lightning Experience Readiness Check some of your organization's data, including setup and configuration data, may be saved and/or processed by Heroku and/or a third party, who may offer different privacy and security protections for such data. Salesforce is not responsible for the privacy and security of the data that is shared as a result of your decision to enable the Lightning Experience Readiness Check.

We've also put together an enablement pack with sample documents you can download and customize, and it contains a sample gap analysis checklist. You can use this checklist to help guide your assessment.

CHAPTER 4 Your Lightning Experience Rollout Strategy

In this chapter ...

- Enabling Lightning Experience
- Recommended Features for Lightning Experience
- Set Up Users for Lightning Experience
- Turn Lightning Experience On!
- Switching Between Lightning Experience and Salesforce Classic

One of the best investments you can make in your Lightning Experience rollout is to have a clear plan. A plan helps you do things in the right order, identify key resources, communicate with everyone, and have a clear end date in mind.

Depending on the size of your organization, you could be working with a Change Management department or have a project manager assigned to the rollout. Or perhaps you're the one charged with organizing and executing the rollout from start to finish. However the work gets divvied up, use Welcome to Your Lightning Experience Rollout on page 187 to help define your overall rollout strategy.

Enabling Lightning Experience

At this point, you've evaluated the benefits of Lightning Experience and are ready to deploy to some or all of your users. Next, we'll walk you through the technical steps to prepare for and enable Lightning Experience for your users.

Let's dive into the details.

Recommended Features for Lightning Experience

If you haven't already, consider enabling and setting up several supporting features before turning on Lightning Experience. These features ensure that the new user interface is fully optimized to help your sales reps sell faster and smarter.

Related Files

Leverage the rich features and flexibility of Salesforce Files from standard Salesforce objects and your custom objects. Add the new Files related list to page layouts so that users can upload files to records, see files associated with records, and quickly page through files in the modern, visually rich preview player.

Enhanced Notes

Help your users stay organized with our enhanced note-taking tool. It's a breeze to create rich-text notes and quickly relate them to specific records. When you turn on Notes, remember to add the new Notes related list to page layouts so that your users can create and read notes directly from their records.

Path

Help your users adopt your company's processes to work more efficiently or close more deals. Set up a process based on an object's status or stage. Add guidance for success at each step in the process.

Duplicate Management

Create duplicate rules to alert users if they're about to create a duplicate record. Craft your duplicate rules to control whether and when users can create duplicate records in Salesforce.

Lead Conversion

Keep the sales process moving by letting your sales reps convert qualified leads to contacts. At the time of conversion, sales reps select an account for the newly created contact, or create an account if it's not yet in Salesforce. Reps can also create an opportunity on the spot. Add the Convert button to the Lead page layout.

News

Give your reps instant access to relevant, timely news about their accounts, contacts, leads, and opportunities. This feature is available only in English, so if your org uses a different primary language, you might not want this option.

Social Accounts, Contacts, and Leads

Keep the team up to date by letting reps link their accounts, contacts, and leads to matching Twitter profiles. Users can see Twitter user profiles and people in common in Salesforce and quickly access tweets.

Shared Activities

Represent activity relationships more accurately by letting your sales reps relate multiple contacts to individual events and tasks. Shared Activities is forever. After the feature's enabled, it can't be disabled.

My Domain

Create a custom domain with the Salesforce My Domain feature. A custom domain helps you manage login and authentication. You must set up a custom domain to use Lightning components in Lightning component tabs, Lightning Pages, the Lightning App Builder, and standalone apps. To set up a custom domain, see My Domain in the Salesforce Help.

We review some of these features in more depth later on.

You can enable some of these recommended features from the Lightning Experience Migration Assistant, which you can find in Salesforce Classic Setup by clicking Lightning Experience.

Set Up Users for Lightning Experience

Next up, make sure that the desired users get access to Lightning Experience. By default, the "Lightning Experience User" permission is automatically enabled for all users with a standard Salesforce profile. But you can fine-tune access to Lightning Experience with custom profiles or permission sets. Meaning you can do a limited rollout to a pilot group or enable a specific team of users who can benefit from the new interface. Or go for it and set your entire organization loose. The power is yours.

Note: Okay, if you use Group Edition, that's overstating things. You can't remove the "Lightning Experience User" permission from your org's standard profiles, and custom profiles aren't available. So it's "all or nothing" if you want to turn on Lightning Experience in Group Edition—all your users will be enabled to use the new interface. But you can control who is immediately switched into the new interface and who remains in the classic interface until they're ready to switch themselves.

First things first, do you have users with standard profiles who shouldn't get access to Lightning Experience yet? Move these users to custom profiles that don't include the "Lightning Experience User" permission.

For users with custom profiles, decide who gets Lightning Experience. Custom profiles don't automatically include the "Lightning Experience User" permission so it's up to you. Consider these options.

- Want to test Lightning Experience with a small group of custom profile users but you don't want to turn it on for everyone assigned to these profiles? Create a permission set that includes the "Lightning Experience User" permission. Then apply the permission set to each pilot user. When you turn on Lightning Experience, only pilot users see the new interface.

  Are you new to permission sets? Check out this walkthrough! Walk Through It: Create, Edit, and Assign a Permission Set

- Interested in rolling out Lightning Experience to specific custom profiles? Or ready to enable all your custom profile users? Adding the "Lightning Experience User" permission to profiles is the fastest way to mass-enable the new user interface.

  If you're dealing with many profiles, you can tackle them at the same time with the Data Loader. But remember, with great power comes great responsibility. Take care you don't inadvertently enable or disable other features.

If you're going to limit Lightning Experience to a subset of your users, we recommend that you keep all members of a functional team on the same experience. If you have team members who often share links and work closely together, include them all in your pilot. See Gotchas If Users Switch Between Lightning Experience and Salesforce Classic for more details.

Decide Who Immediately Switches to Lightning Experience

The last step is to decide which of your users should immediately switch to the new interface when you enable Lightning Experience and who should stay in Salesforce Classic. You can choose what's right for each user. If your team isn't fully trained on Lightning Experience yet or it's not a good time to introduce a major change, you can opt to leave everyone in Salesforce Classic when Lightning Experience is turned on.

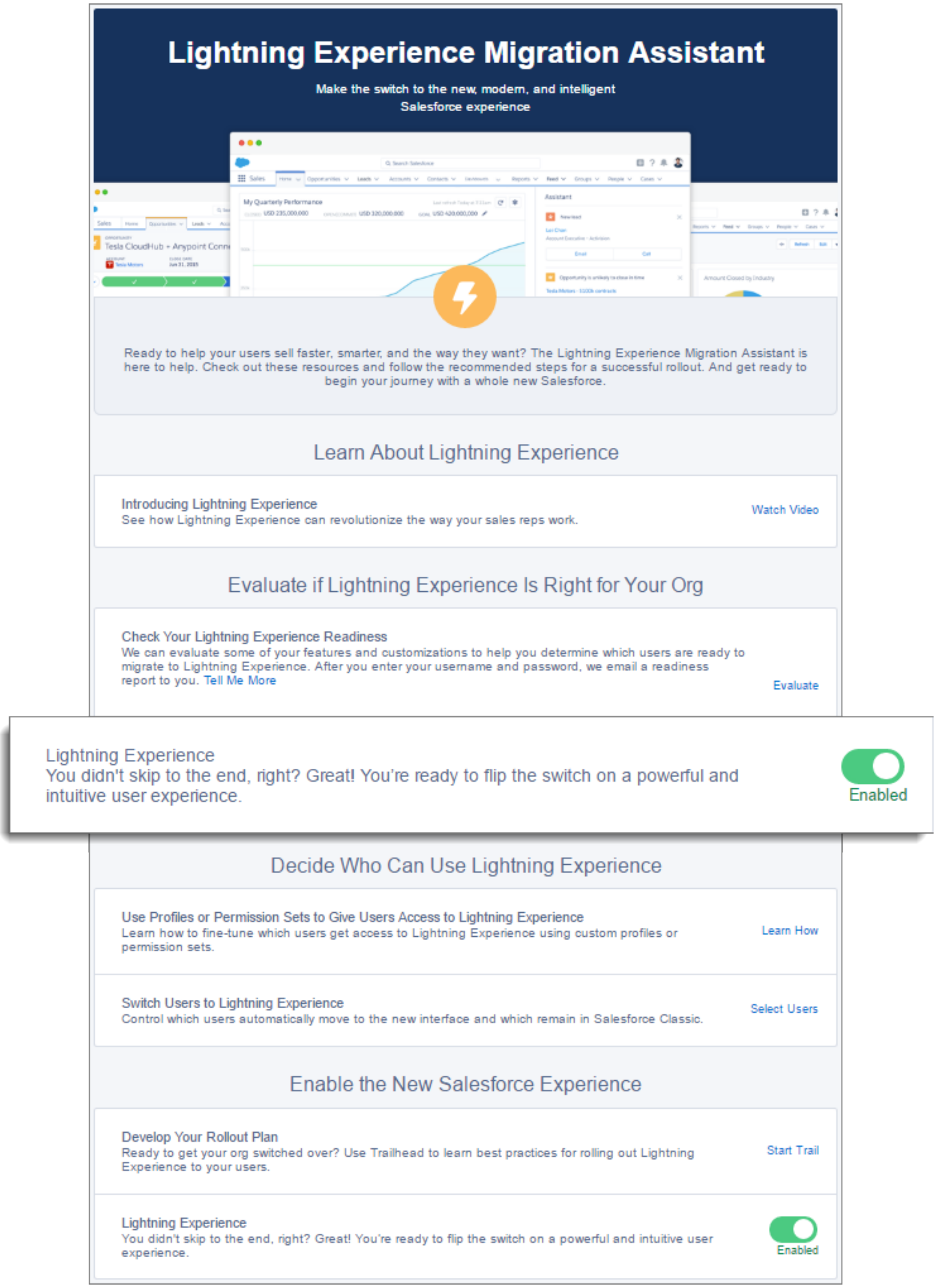
Tip: Remember, everyone who's enabled for Lightning Experience automatically gets the Switcher. Meaning users who remain in Salesforce Classic can switch themselves to Lightning Experience whenever they're ready. And of course, users assigned to the new interface can switch back to Salesforce Classic as needed.

To assign the right interfaces for your users, use the Lightning Experience Migration Assistant. From Setup in Salesforce Classic, click **Lightning Experience**. In the "Decide Who Can Use Lightning Experience" section, find the "Switch Users to Lightning Experience" item and click **Select Users**.

Turn Lightning Experience On!

You've knocked off the “Learn” and “Launch” items in your rollout plan. You've set up the Salesforce features that optimize the new user interface. You've enabled the right users and designated who gets the new interface right away and who stays in the classic interface. You're ready to go live! Fortunately, it’s easy to turn on Lightning Experience using the Migration Assistant.

From Setup in Salesforce Classic, click **Lightning Experience**. On the Lightning Experience Migration Assistant page, click the **Lightning Experience** button to set it to **Enable**.



Lightning Experience Migration Assistant

Make the switch to the new, modern, and intelligent Salesforce experience

Ready to help your users sell faster, smarter, and the way they want? The Lightning Experience Migration Assistant is here to help. Check out these resources and follow the recommended steps for a successful rollout. And get ready to begin your journey with a whole new Salesforce.

Learn About Lightning Experience

Introducing Lightning Experience
See how Lightning Experience can revolutionize the way your sales reps work.

Evaluate if Lightning Experience Is Right for Your Org

Check Your Lightning Experience Readiness
We can evaluate some of your features and customizations to help you determine which users are ready to migrate to Lightning Experience. After you enter your username and password, we email a readiness report to you. Tell Me More

Lightning Experience
You didn't skip to the end, right? Great! You're ready to flip the switch on a powerful and intuitive user experience.

Decide Who Can Use Lightning Experience

Use Profiles or Permission Sets to Give Users Access to Lightning Experience
Learn how to fine-tune which users get access to Lightning Experience using custom profiles or permission sets.

Switch Users to Lightning Experience
Control which users automatically move to the new interface and which remain in Salesforce Classic.

Enable the New Salesforce Experience

Develop Your Rollout Plan
Ready to get your org switched over? Use Trailhead to learn best practices for rolling out Lightning Experience to your users.

Lightning Experience
You didn't skip to the end, right? Great! You're ready to flip the switch on a powerful and intuitive user experience.

That's it! The users you've enabled and set up to immediately switch to the new interface automatically start enjoying Lightning Experience when their current session refreshes or when they log in. The users you've enabled but opted to leave in Salesforce Classic now have access to the Switcher and can switch to Lightning Experience whenever they want.

Note: Remember that for all editions other than Group Edition, you can set up access for specific users via permission sets or custom profiles.

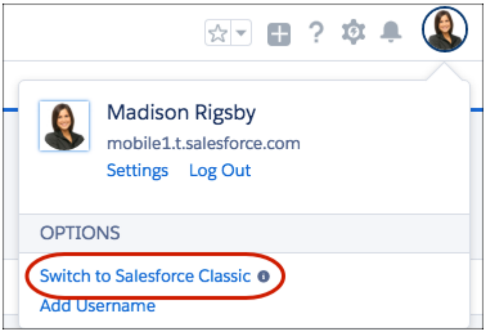
Switching Between Lightning Experience and Salesforce Classic

Meet the Switcher. When Lightning Experience is enabled, you can use this feature to switch back and forth between Lightning Experience and Salesforce Classic.

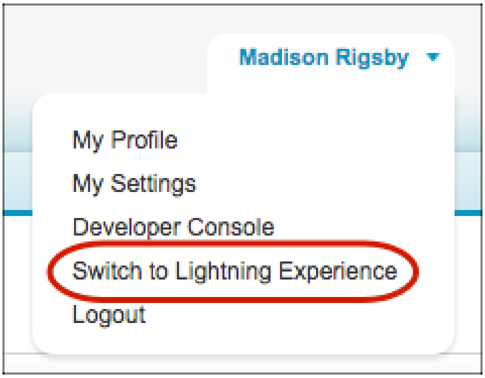
The Switcher is smart. Whenever you switch, it remembers that user experience as your default preference. So if you switch to Salesforce Classic, the classic interface becomes your default preference. If you switch back to Lightning Experience, the new interface becomes your default preference. And so on and so on.

How do you get to the Switcher?

In the Lightning Experience header, click the profile picture and select **Switch to Salesforce Classic**.



To find the Switcher in Salesforce Classic, click the name in the upper-right corner, then select **Switch to Lightning Experience**.



A Few Switcher Gotchas

Switching between Lightning Experience and Salesforce Classic is fast and easy. But switching between interfaces affects the underlying URL routing logic and can lead to some unexpected results when links are resolved. This means that there can be snags with bookmarks and sharing links in emails or Chatter posts, especially if your organization has a subset of users authorized for Lightning Experience while others are still using Salesforce Classic.

If a user clicks a link to something that's not supported in Lightning Experience, Salesforce switches to Salesforce Classic in a new tab while the original Lightning Experience window stays open.

Users who aren't enabled for Lightning Experience can't access Lightning Experience links, including links posted to Chatter or emailed from a Lightning Experience user.

Tip: Keep all members of a functional team on the same experience. If you have team members who often share links and work closely together, include them all in the pilot. You can roll out Lightning Experience to a specific set of users using profiles or permission sets.

CHAPTER 5 Navigation and Setup in Lightning Experience

In this chapter ...

- The Lightning Experience Navigation Bar
- The App Launcher in Lightning Experience
- Meet the New and Improved Setup
- Object Manager
- Search for Records
- Lightning Experience Help Menu

Setup is where you make the magic happen. As a Salesforce admin or developer, you spend a lot of time using Setup. It's where you customize and configure your organization, support users, build functionality, and more.

One of the huge productivity upgrades that comes with the new Lightning Experience is the improved Setup. We've done a lot of usability testing and refactoring to revamp the Setup tree. We simplified it to have a logical and easy-to-navigate structure, using broad categories to make things more discoverable. In addition, child nodes are now in alphabetical order.

The navigation bar in Lightning Experience provides an efficient and consistent interface to navigate through your organization's various apps and items. Similar to Salesforce Classic, apps in Lightning Experience give your users access to sets of objects, tabs, and other items all in one convenient bundle in the navigation bar. However, apps in Lightning Experience take things to another level beyond apps in Salesforce Classic by letting you brand your apps with a custom color and logo. In Lightning Experience you can even include Lightning page tabs and a utility bar that allows instant access to productivity tools, like integrated voice, in the footer of Lightning Experience.

Ready to see them in action?

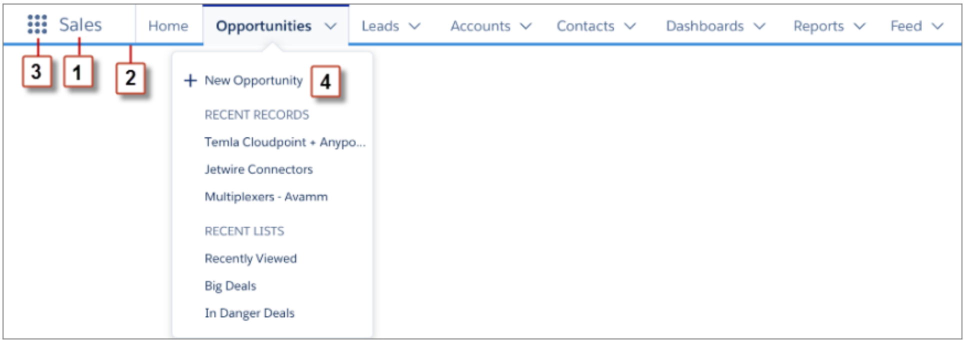
The Lightning Experience Navigation Bar

If you know Salesforce Classic, the Lightning Experience navigation model will feel like a familiar friend, only better.

Each Lightning app has a navigation bar at the top of the page, letting your users:

- Find what they need using item names for easy recognition
- Complete actions and access recent records and lists with a single click

Think of the navigation bar as a container for a set of items and functionality. It's always there, but the items within it change based on the app you're using.



+ New Opportunity

RECENT RECORDS

Temla Cloudpoint + Anypo...

Jetwire Connectors

Multiplexers - Avamm

RECENT LISTS

Recently Viewed

Big Deals

In Danger Deals

- The app name displays on the left side of the navigation bar (1) and custom colors and branding (2) make each app unique and easy to identify.
- Your users can access other items and apps by clicking the App Launcher icon (3) or the app name.
- Your users can create records and access recent records and lists directly from the navigation bar (4) for items like Opportunities.

What can you put in the Lightning app navigation bar?

If you're familiar with Salesforce Classic, you know that Classic apps can contain:

- Most standard objects, including Home, the main Chatter feed, Groups, and People
- Your org's custom objects
- Visualforce tabs
- Lightning component tabs
- Canvas apps via Visualforce tabs

- Web tabs

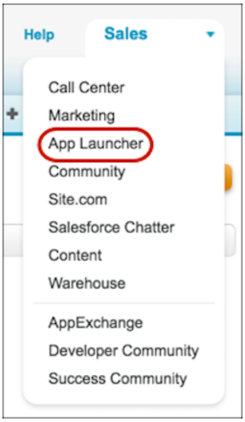
Lightning apps can contain everything on this list plus Lightning Page tabs and utilities like Lightning Voice. If your org uses utility features, you can enable a utility bar in your app that allows instant access to productivity tools, like integrated voice, in the Lightning Experience footer.

As we mentioned, you and your users can still get to custom apps and objects via the App Launcher, which we'll look at next.

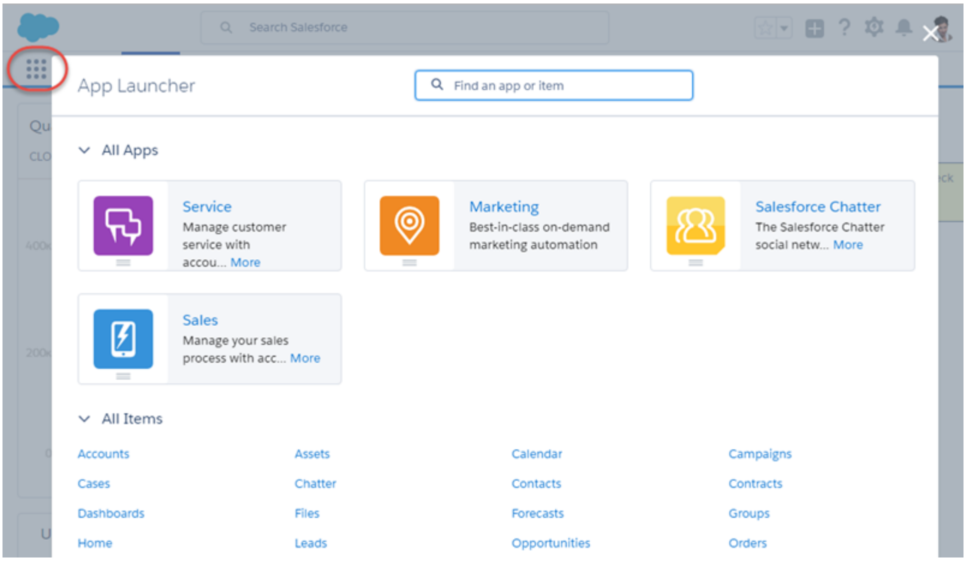
The App Launcher in Lightning Experience

In Lightning Experience, your users switch between apps through the App Launcher. They can browse the App Launcher to find available apps. Or they can search for an app by name from the `Find an app or item` box at the top of the App Launcher.

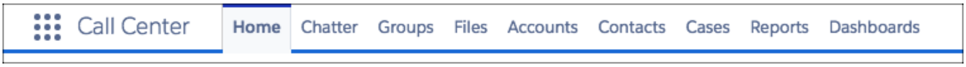
In Salesforce Classic, your users commonly switch between apps through the Force.com app menu or the App Launcher.



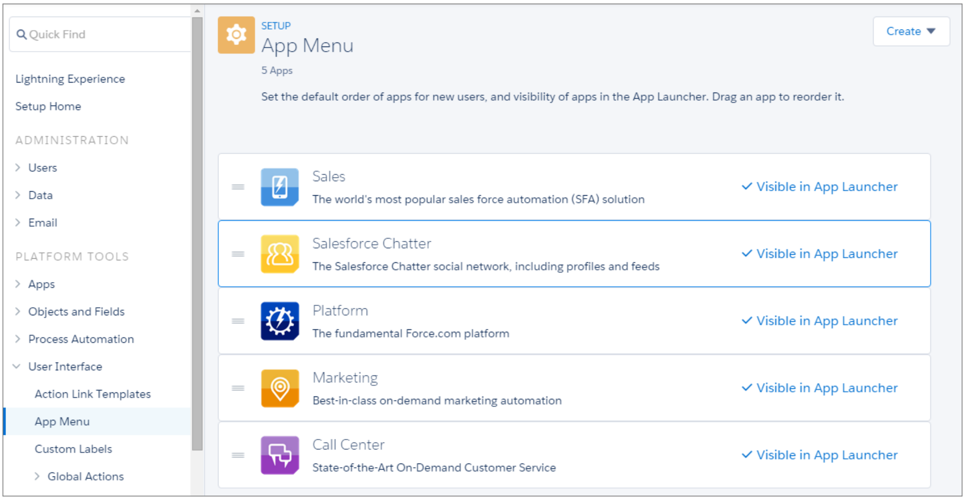
To get to the Lightning Experience App Launcher, click ⋮⋮⋮ from any page. Apps show up as large tiles under All Apps. Apps can include Salesforce standard apps, custom apps, and connected apps like Gmail and Google Drive. Other items, such as custom objects, tasks, events, and the feed, show up under All Items.



When you click in an app, you see the app name on the left side of the navigation bar. Items associated with the app appear to the right. Your users can create records and access recent records directly from the navigation bar for certain items like Accounts.



As a Salesforce admin, you can change which apps appear on the Lightning Experience App Launcher. You can also control the order in which the apps appear from the app menu.



Quick Find

Lightning Experience

Setup Home

ADMINISTRATION

› Users

› Data

› Email

PLATFORM TOOLS

› Apps

› Objects and Fields

› Process Automation

⌄ User Interface

Action Link Templates

App Menu

Custom Labels

› Global Actions

SETUP

App Menu

5 Apps

Create ▼

Set the default order of apps for new users, and visibility of apps in the App Launcher. Drag an app to reorder it.

Sales — The world's most popular sales force automation (SFA) solution — ✓ Visible in App Launcher

Salesforce Chatter — The Salesforce Chatter social network, including profiles and feeds — ✓ Visible in App Launcher

Platform — The fundamental Force.com platform — ✓ Visible in App Launcher

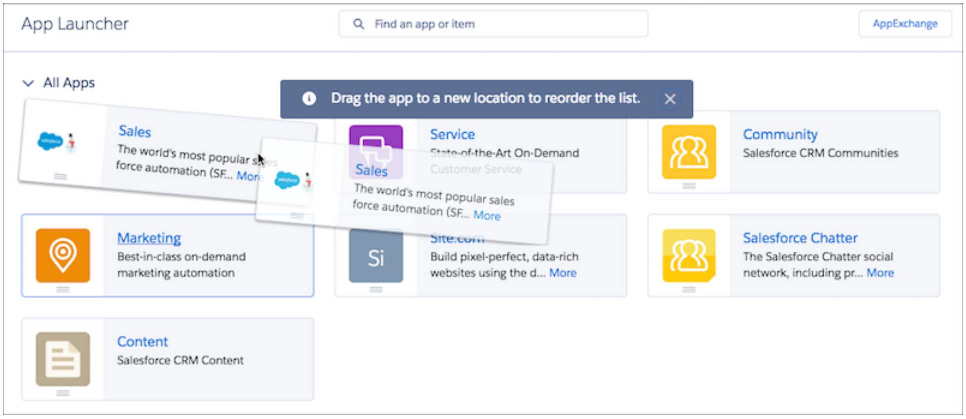
Marketing — Best-in-class on-demand marketing automation — ✓ Visible in App Launcher

Call Center — State-of-the-Art On-Demand Customer Service — ✓ Visible in App Launcher

1. From Setup, enter `App Menu` in the `Quick Find` box, then select **App Menu**.
2. From the list of app menu items, drag the apps to change their order. Changes take effect immediately.
3. Optionally, click **Visible in App Launcher** or **Hidden in App Launcher** to show or hide individual apps from the App Launcher for all users in the org.

The app menu lists all apps installed in the org. However, the apps that users see in their App Launcher and app menu vary depending on each app's visibility settings and user permissions. Users see only the apps that they are authorized to see according to their profile or permission sets.

Users can drag tiles to sort their personal view of the App Launcher to their liking.



App Launcher

Find an app or item

AppExchange

⌄ All Apps

Drag the app to a new location to reorder the list. ✕

Sales — The world's most popular sales force automation (SF... More

Service — State-of-the-Art On-Demand Customer Service

Sales — The world's most popular sales force automation (SF... More

Community — Salesforce CRM Communities

Marketing — Best-in-class on-demand marketing automation

Site.com — Build pixel-perfect, data-rich websites using the d... More

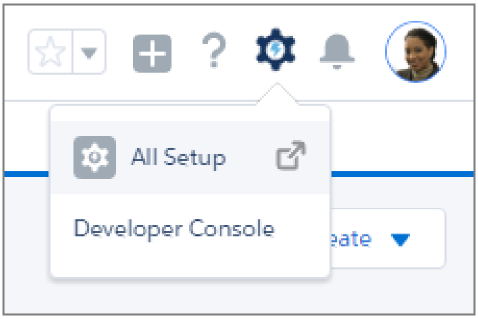
Salesforce Chatter — The Salesforce Chatter social network, including pr... More

Content — Salesforce CRM Content

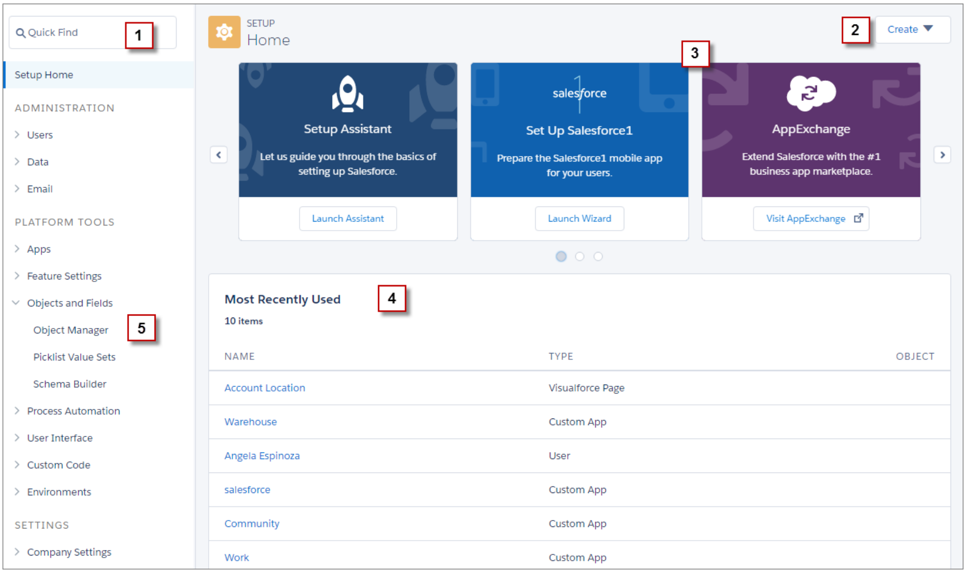
Meet the New and Improved Setup

A nip here? A tuck there? No, we've given Setup a whole new face!

You can navigate to Setup from the top of any page in Lightning Experience by clicking ⚙ > **All Setup**.



The Setup tree has been completely reorganized and recategorized. In Salesforce Classic, the Setup tree had a lot of specific node categories, often with several nodes as nested subcategories. Sometimes there were many different ways to get to a destination. You might have mastered the click paths after a while, but to new users, this structure was often an overwhelming hurdle. The new Setup provides a streamlined interface for viewing and managing your administrative setup tasks.



Setup Assistant

Let us guide you through the basics of setting up Salesforce.

Launch Assistant

Set Up Salesforce1

Prepare the Salesforce1 mobile app for your users.

Launch Wizard

AppExchange

Extend Salesforce with the #1 business app marketplace.

Visit AppExchange

Most Recently Used

10 items

NAME	TYPE	OBJECT
Account Location	Visualforce Page	
Warehouse	Custom App	
Angela Espinoza	User	
salesforce	Custom App	
Community	Custom App	
Work	Custom App	

The new Setup includes these enhancements.

- The Quick Find (1) lets you quickly navigate to any node using a keyword. Quick Find is the best way to find what you're looking for if you know its name. Quick Find is your power tool for getting where you need to go!
- The Create menu (2) gives you quick access to common Setup creation functions—including users, custom objects, custom tabs, apps, email templates, and processes—without having to drill down through the Setup tree to get the page. You can get to the Create menu from any page in Setup.
- A carousel of quick-access tiles (3) gives you instant access to important setup tools and information, as well as the release notes. The Lightning Experience tile and Setup Salesforce1 tile help you enable your company for the new and improved UI, and mobile data access. There's also a link to download SalesforceA—which lets you do Salesforce administration from a mobile app—and a link to the System Status screen so you can view your org's performance and usage data.
- The Most Recently Used list (4) on the Setup Home page shows your most recently used records or customization features in Setup. You can quickly link back to what you were working on by clicking its name.
- The Object Manager (5) provides a one-stop shop for managing all objects in your organization, both standard and custom. We'll look at the object manager in more detail shortly.

Users can access their personal settings at the top of any page by clicking their profile image, then clicking **Settings**.

Administration, Platform Tools, and Settings, Oh My!

In the improved Setup, we've changed the five Setup tree sections from Administer, Build, Deploy, and Checkout to three sections: Administration, Platform Tools, and Settings. We completely reorganized all the child nodes to fit into these sections where appropriate, and added broader subcategories to make finding nodes easier, even if you don't know the exact name you're looking for.

Not only that, there are fewer child nodes to choose from, as many repetitive nodes in the tree have been removed. We reorganized the platform tools into a more process-oriented organizational structure rather than being feature-oriented. Now you can see pieces of the application life cycle broken up into subcategories: Apps, Objects and Fields, Process Automation, User Interface, Custom Code, and Environments.

At the bottom of the tree, in the Settings section, you can view company information or configure security.

Note: As you're getting familiar with the Setup area, it's important to keep in mind that navigating through Setup is not about memorizing click paths; it's about understanding what you're looking for in order to get to your destination. Depending on your users' profile and permissions, one user might see a different set of items in Setup than another. As a System Administrator, however, you see everything.

Where Did Some of the Nodes Go?

You might have noticed that some of the nodes are missing altogether. Only nodes that are related to customizing the new Lightning Experience are included in the Setup tree. For example, the Service Cloud related nodes are gone, as well as some of the Sales Cloud features. And, we moved all the object-related nodes to the Object Manager, which we'll look at shortly.

Don't worry! From a customization and development standpoint, all of the tools are still there. And as the other features become supported in Lightning Experience, you'll see them in Setup too.

Five Things You Shouldn't Miss in the Improved Setup

Setup Lightning Experience	• Your one stop for Lightning Experience customizations • Learn best practices • Enable/disable Lightning Experience and new features

	- Setup permissions to give users access to Lightning Experience
Create Menu	- On every page in Setup - Quick access to create common items
Object Manager	- All standard and custom objects now live in the Object Manager - All objects now have a standard detail page - You can filter the list of objects and also filter the contents of the detail page to find things quickly
App Menu	Use it to: - Reorder apps in the App Launcher - Make apps visible or invisible in the App Launcher
View Release Notes	- Links to the most recent version of the release notes - Great point of reference for new and existing features

Limitations

Advanced Setup Search isn't available in Lightning Experience. However, you can work around that while in Setup by entering a term in global search and selecting the **in Setup** option in instant results. The search results page lists records that match your search term.

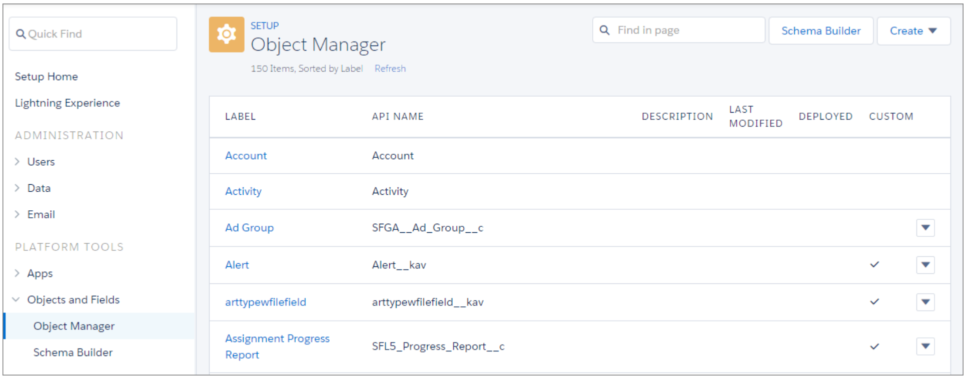
The Setup tree in Lightning Experience is limited to:

- Pages that support Lightning Experience features
- Administration pages that apply across your organization, such as user management, security, and company settings

Use Salesforce Classic to access administration pages for features that aren't in Lightning Experience.

Object Manager

The Object Manager is a one-stop shop for managing all objects in your organization, both standard and custom. Access all objects and their related functions—fields, validation rules, page layouts, and so on—from a single entry point.

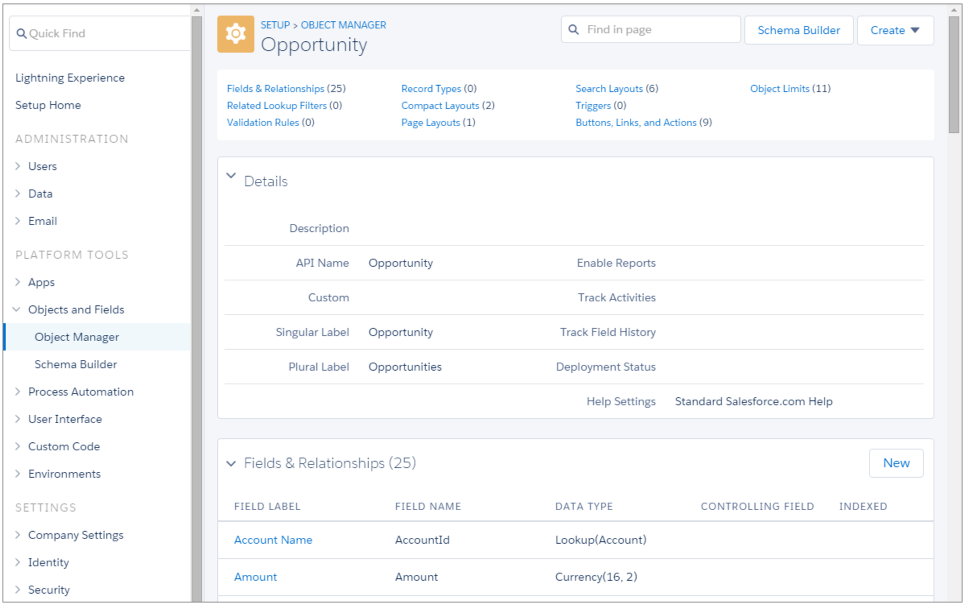


LABEL	API NAME	DESCRIPTION	LAST MODIFIED	DEPLOYED	CUSTOM	
Account	Account					
Activity	Activity					
Ad Group	SFGA__Ad_Group__c					▼
Alert	Alert__kav				✓	▼
arttypewfilefield	arttypewfilefield__kav				✓	▼
Assignment Progress Report	SFL5_Progress_Report__c				✓	▼

To access the Object Manager, from Setup, enter `Object Manager` in the `Quick Find` box, then select **Object Manager**.

- To find an object, enter the first few characters of its label or name in the `Find in page` box.
- To edit a custom object, click ▼ , then **Edit**.
- To view more details about an object or to access its related functions, click the object label.

From the object detail page, you can view the object details and access all related functions, such as fields, validation rules, and page layouts.



Description

API Name Opportunity

Enable Reports

Custom

Track Activities

Singular Label Opportunity

Track Field History

Plural Label Opportunities

Deployment Status

Help Settings Standard Salesforce.com Help

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Account Name	AccountId	Lookup(Account)		
Amount	Amount	Currency(16, 2)		

- To quickly jump to a function or control, use the links at the top of the page.
- To find a function or control by name, enter it in the `Find in page` box.

Limitations

The Object Manager is limited to objects that support Lightning Experience features.

These object functions aren't listed in the Object Manager. You can access them from elsewhere in Setup.

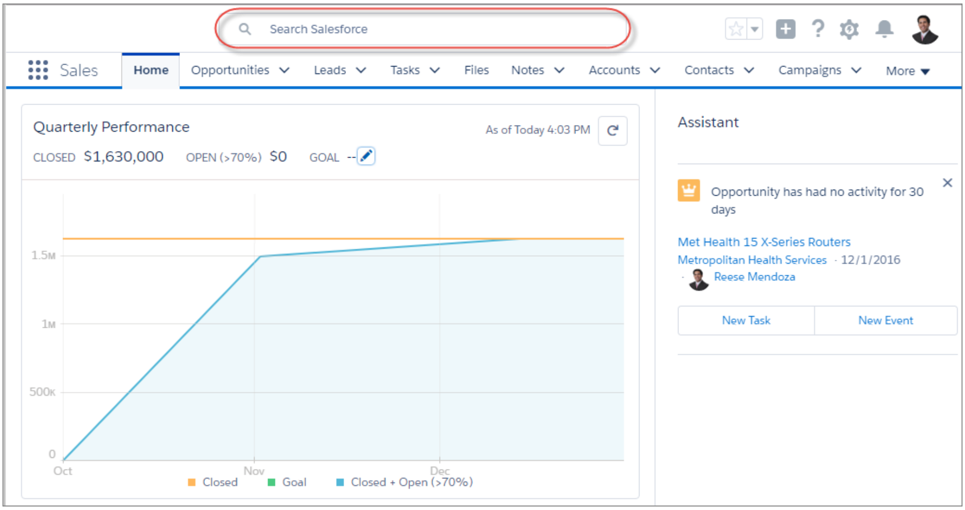
- Case Comment Triggers
- Feed Comment Triggers
- Feed Item Triggers
- Feed Item Layouts
- Group Layouts
- Group Triggers
- Group Member Triggers
- Group Record Triggers
- Publisher Layouts
- Topic Triggers

- Topic Assignment Triggers

Search for Records

You've got a ton of useful data in Salesforce. How do you get to what you need, when you need it? Let's face it, no one has time to browse anymore. It's all about search. So let us introduce you to the Lightning Experience search box. It's at the top of every page, and it's the fastest way to bring what you need right to your fingertips.

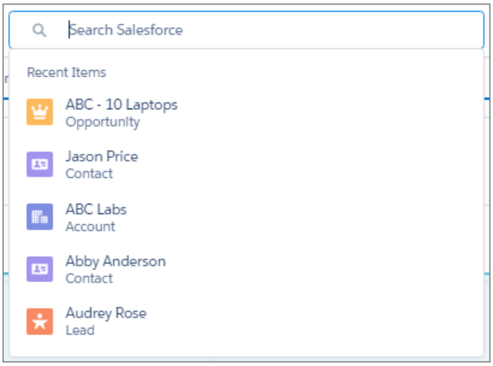
Think of search as your personal assistant—your very smart personal assistant. It anticipates your needs and helps you find what you're looking for from anywhere in the app.



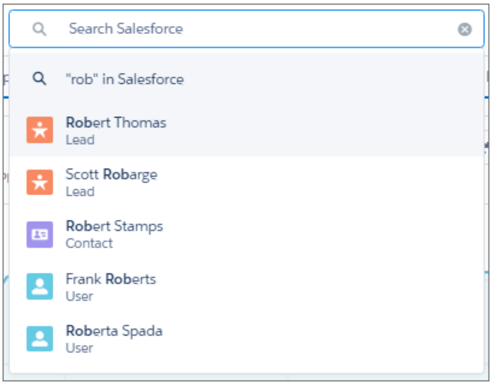
Opportunity has had no activity for 30 days

Met Health 15 X-Series Routers
Metropolitan Health Services · 12/1/2016
· Reese Mendoza

As soon as you click into the box, search starts working for you. A drop-down presents a list of recent items, which you can use as shortcuts to your top-of-mind records.



Start typing, and the list dynamically updates with potential matches from all searchable objects. If you see what you're looking for, select it to go right to the record.



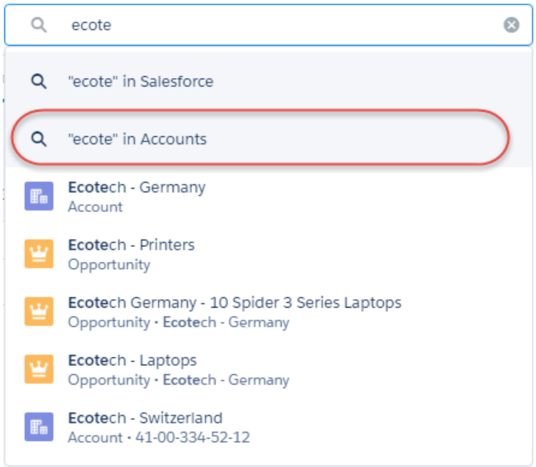
Speaking of typing, let's take a moment to talk about *what* you're typing. While search is expertly trained to find what you're looking for, you can help it help you by being as specific as possible when you search.

What does that mean, specifically? First, enter as many terms as you know. Let's say you're looking for *John Smith* from your sales team. Enter his full name. If you're looking for the sales report from March 2015, enter **sales report March 2015**.

Second, even if you're not completely sure what you're looking for, use your best guess. This is where search gets to shine. For example, spell correction kicks in automatically when there aren't any results for your initial search term. Synonyms of your search term are also returned. And if you search for laptop, you also get results for laptops.

Note: Operators (like AND, OR, and AND NOT), exact phrase searches (terms surrounded in " "), and wildcards (like * and ?) affect the efficiency and effectiveness of the search engine. Although they're available, use wildcards, operators, and exact phrase searches only if you're not finding what you're looking for using basic keyword searches.

OK, so you've entered your search terms. Don't see what you need in the drop-down? Don't worry—you've got options. To see results only for the object you're on, select the second option in the instant results drop-down.



ecote

"ecote" in Salesforce

"ecote" in Accounts

Ecotech - Germany
Account

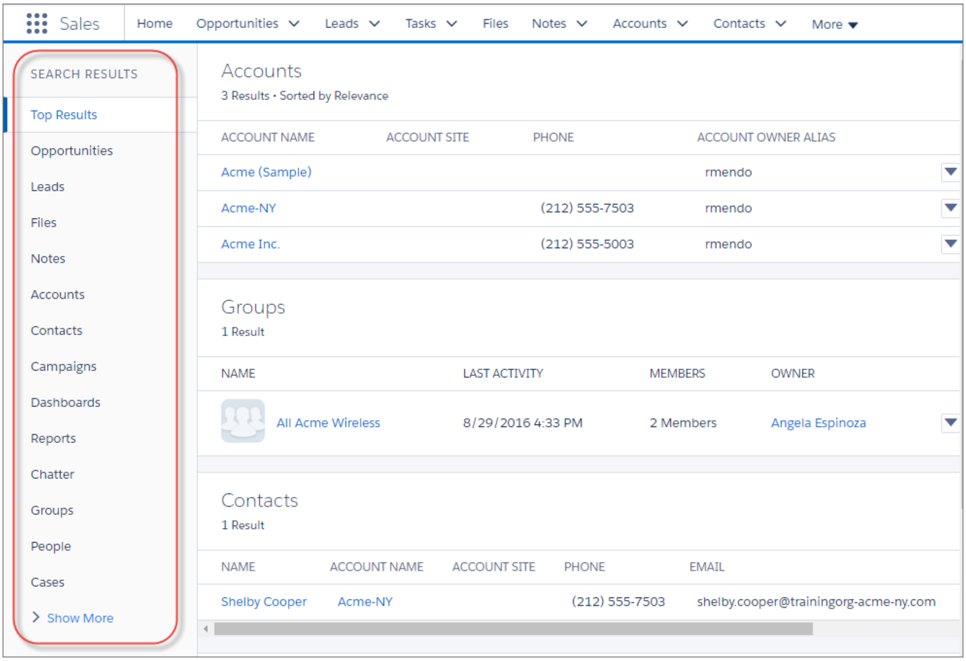
Ecotech - Printers
Opportunity

Ecotech Germany - 10 Spider 3 Series Laptops
Opportunity • Ecotech - Germany

Ecotech - Laptops
Opportunity • Ecotech - Germany

Ecotech - Switzerland
Account • 41-00-334-52-12

Alternatively, you can select the first option or press Enter to search across your entire organization. You'll land on the Top Results page, which shows you the most relevant results from your most frequently used objects.



Accounts

3 Results • Sorted by Relevance

ACCOUNT NAME	ACCOUNT SITE	PHONE	ACCOUNT OWNER ALIAS
Acme (Sample)			rmendo
Acme-NY		(212) 555-7503	rmendo
Acme Inc.		(212) 555-5003	rmendo

Groups

1 Result

NAME	LAST ACTIVITY	MEMBERS	OWNER
All Acme Wireless	8/29/2016 4:33 PM	2 Members	Angela Espinoza

Contacts

1 Result

NAME	ACCOUNT NAME	ACCOUNT SITE	PHONE	EMAIL
Shelby Cooper	Acme-NY		(212) 555-7503	shelby.cooper@trainingorg-acme-ny.com

From this high-level overview, it's easy to home in on what you need. See results for a specific object by selecting it on the left, under Search Results. Searchable objects are listed in the same order they appear in the global navigation bar.

If you don't see an object you need under Search Results, don't worry, it's close at hand. Select **Show More** to see all objects available to you, listed in alphabetical order.

Search results are sorted by relevance. That sounds smart—and it is!—but what does it mean, exactly? Several things are considered, like how unique the search term is, how often the search term appears in a record, and whether an item is owned by the person searching. You can also sort search results by other criteria, like record name, date added, and so on. Click column headers or use the sort drop-down. (If the sort button is disabled, it's because the search results layout doesn't contain any fields that are sortable.)

What To Do If You Don't Find What You're Searching For

- Check your spelling and verify that you entered the full search term.
- Check whether the object or field is searchable.
- Make sure you have access to the record. Search only returns results you have permission to view.

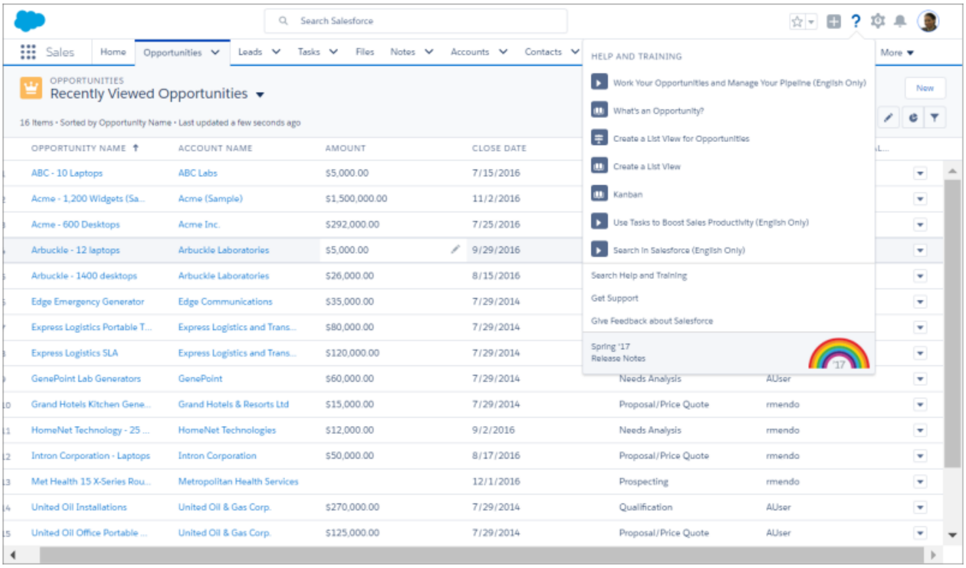
- If you recently created or updated the record, wait a few minutes for the record to be indexed. If you can't find your record after 15 minutes, contact your admin.

We think you'll agree you've found the perfect assistant—it has a broad perspective, yet directs you to the important things. It also doesn't take vacations or ask for a raise. It saves you so much time and makes getting around Salesforce so easy, though, that it probably deserves one.

Lightning Experience Help Menu

In Lightning Experience, each page has a contextual help menu with links to resources—help topics, walkthroughs, videos, developer guides, and PDFs—related to the specific tasks on that page.

On object pages, the menu replaces the "Help for this Page" link that you're used to seeing in Salesforce Classic. The "Help for this Page" link still appears on many Setup pages, but we encourage you to use the help menu instead for a broader selection of useful information.



OPPORTUNITIES
Recently Viewed Opportunities

16 Items • Sorted by Opportunity Name • Last updated a few seconds ago

OPPORTUNITY NAME	ACCOUNT NAME	AMOUNT	CLOSE DATE		
ABC - 10 Laptops	ABC Labs	$5,000.00	7/15/2016		
Acme - 1,200 Widgets (Sa...	Acme (Sample)	$1,500,000.00	11/2/2016		
Acme - 600 Desktops	Acme Inc.	$292,000.00	7/25/2016		
Arbuckle - 12 laptops	Arbuckle Laboratories	$5,000.00	9/29/2016		
Arbuckle - 1400 desktops	Arbuckle Laboratories	$26,000.00	8/15/2016		
Edge Emergency Generator	Edge Communications	$35,000.00	7/29/2014		
Express Logistics Portable T...	Express Logistics and Trans...	$80,000.00	7/29/2014		
Express Logistics SLA	Express Logistics and Trans...	$120,000.00	7/29/2014		
GenePoint Lab Generators	GenePoint	$60,000.00	7/29/2014	Needs Analysis	AUser
Grand Hotels Kitchen Gene...	Grand Hotels & Resorts Ltd	$15,000.00	7/29/2014	Proposal/Price Quote	rmendo
HomeNet Technology - 25 ...	HomeNet Technologies	$12,000.00	9/2/2016	Needs Analysis	rmendo
Intron Corporation - Laptops	Intron Corporation	$50,000.00	8/17/2016	Proposal/Price Quote	rmendo
Met Health 15 X-Series Rou...	Metropolitan Health Services		12/1/2016	Prospecting	rmendo
United Oil Installations	United Oil & Gas Corp.	$270,000.00	7/29/2014	Qualification	AUser
United Oil Office Portable ...	United Oil & Gas Corp.	$125,000.00	7/29/2014	Proposal/Price Quote	AUser

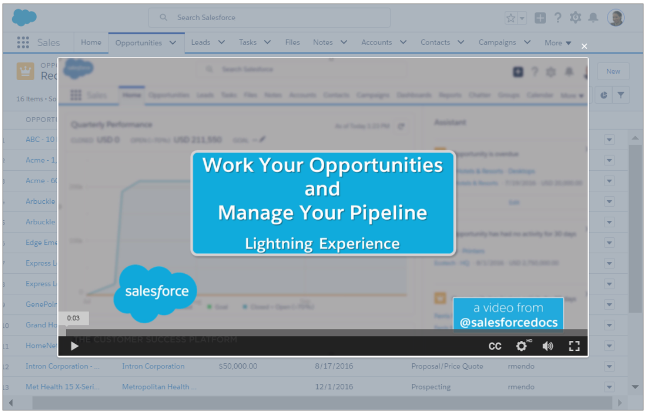
HELP AND TRAINING

- Work Your Opportunities and Manage Your Pipeline (English Only)
- What's an Opportunity?
- Create a List View for Opportunities
- Create a List View
- Kanban
- Use Tasks to Boost Sales Productivity (English Only)
- Search in Salesforce (English Only)

Search Help and Training
Get Support
Give Feedback about Salesforce

Spring '17
Release Notes

When you click a video link (▶), a video player appears, allowing you to watch without leaving Salesforce.



Walkthrough links (☰) take you to the first step of the walkthrough in your organization. Help, developer guide, and PDF links open in a new browser tab.

You can't customize the items in the help menu. If you've created custom help links, those links still work from the "Help for this Page" links in the framed Salesforce Classic pages. However, your custom help links don't appear in the new Help menu.

CHAPTER 6 Opportunities, Leads, and Selling in Lightning Experience

In this chapter ...

- Explore the Opportunity Workspace
- Opportunity Home
- Explore the Lead Workspace

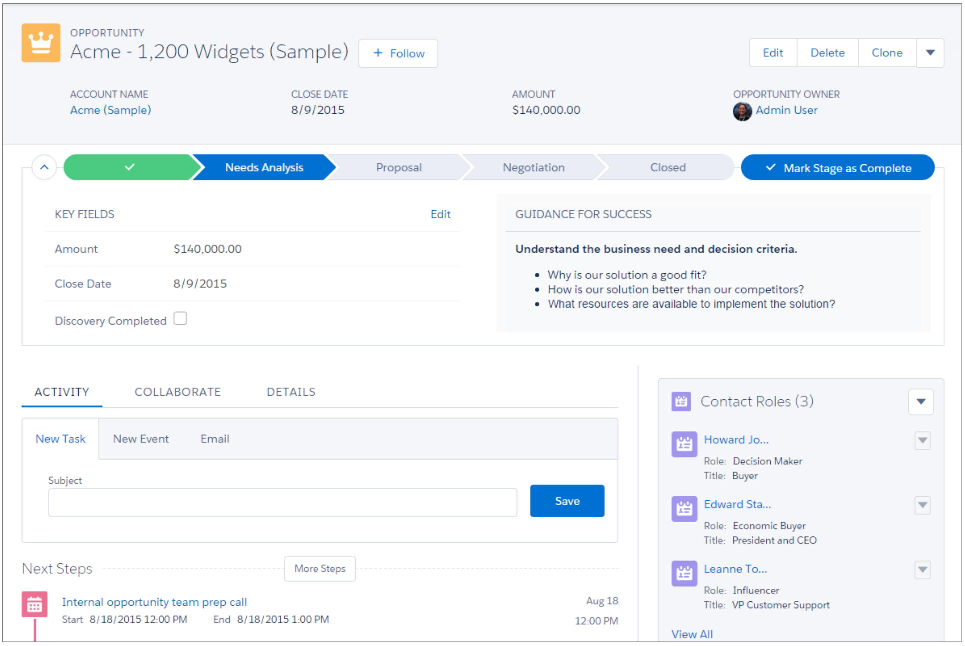
We've taken your sales process and put it into an action-first workspace, designed to help your sales reps work their deals faster and smarter. You can customize coaching scripts for each step in the sales process, create records quickly with fewer clicks, and ultimately close deals faster.

Explore the Opportunity Workspace

Opportunities have gotten a makeover! When you visit an opportunity record in Lightning Experience, you'll see some great enhancements that help your sales users get the most out of their opportunities.

Here are just a few things your users can do from the opportunity workspace:

- Create and update tasks and meetings, log calls, and send email
- View key information for a deal, like its key players, account, close date, amount, owner, and stage
- Update an opportunity's stage, close date, and amount
- View or add members to an opportunity team
- View relevant, timely news about the account the opportunity is related to



ACCOUNT NAME: Acme (Sample)
CLOSE DATE: 8/9/2015
AMOUNT: $140,000.00
OPPORTUNITY OWNER: Admin User

KEY FIELDS	
Amount	$140,000.00
Close Date	8/9/2015
Discovery Completed	□

GUIDANCE FOR SUCCESS

Understand the business need and decision criteria.

- Why is our solution a good fit?
- How is our solution better than our competitors?
- What resources are available to implement the solution?

Next Steps

Internal opportunity team prep call — Aug 18 12:00 PM
Start 8/18/2015 12:00 PM End 8/18/2015 1:00 PM

Contact Roles (3)

Howard Jo...
Role: Decision Maker
Title: Buyer

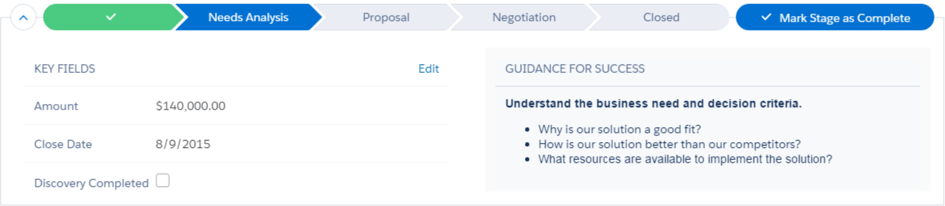
Edward Sta...
Role: Economic Buyer
Title: President and CEO

Leanne To...
Role: Influencer
Title: VP Customer Support

You don't have to do anything to get the opportunity workspace working for your users. However, as an administrator, you can enhance your sales users' workflow by customizing the sales path and the activity timeline to their needs.

Path

Paths guide your sales users through each stage of your company's sales process. Paths help users stay focused on important tasks so they can close deals or complete work quickly.



KEY FIELDS Edit

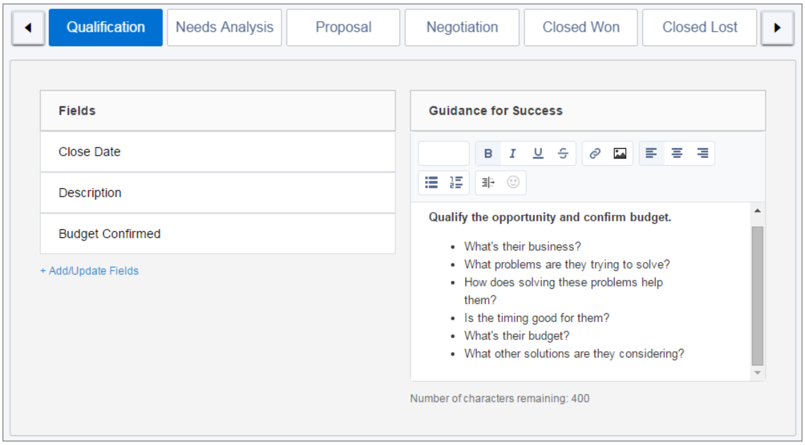
Amount	$140,000.00
Close Date	8/9/2015
Discovery Completed	□

GUIDANCE FOR SUCCESS

Understand the business need and decision criteria.

- Why is our solution a good fit?
- How is our solution better than our competitors?
- What resources are available to implement the solution?

From Setup, enter `Path Settings` in the `Quick Find` box, then select **Path Settings**. You can create a unique path for each record type for leads, opportunities, quotes, and custom objects.



Fields

- Close Date
- Description
- Budget Confirmed

+ Add/Update Fields

Guidance for Success

Qualify the opportunity and confirm budget.

- What's their business?
- What problems are they trying to solve?
- How does solving these problems help them?
- Is the timing good for them?
- What's their budget?
- What other solutions are they considering?

Number of characters remaining: 400

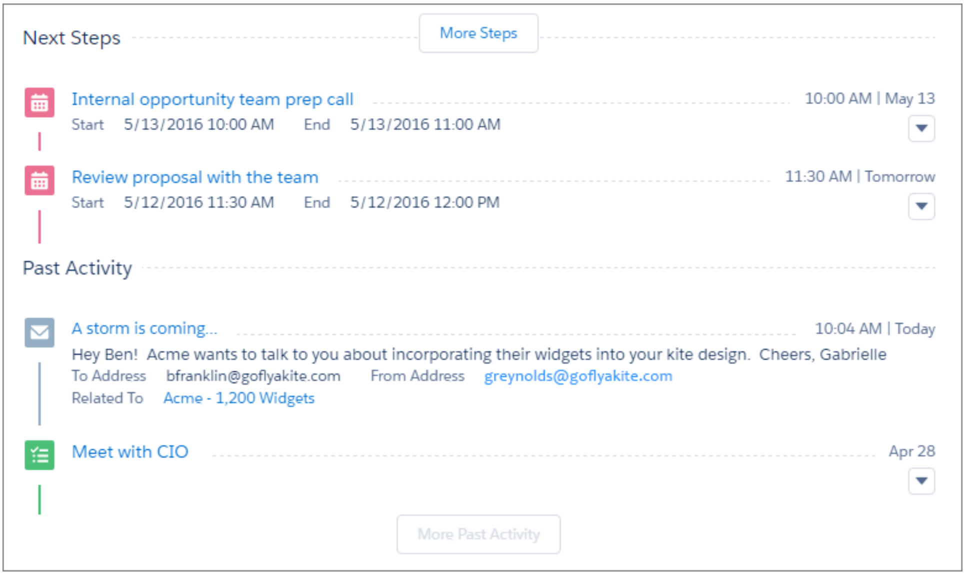
Path Best Practices

- Provide guidance for success content, like links to Chatter posts and videos, tips, or policy reminders—anything that can help sales reps get closer to sealing the deal.
- Keep your system performance optimal by creating sales paths that have 20 or fewer stages.
- Consider labeling sales paths for regions or industries, like "Steel Industry Sales Path."

- If you set up record types, you can have one path for each record type. For example, in the record type New Business, include more prospecting-related fields, but in the record type Existing Customer, include a field or stage for renewals.

Activity Timeline

With the activity timeline, your sales reps can keep a finger on the pulse of their deals. The timeline tracks meetings, tasks, calls, and emails. Reps can see what they've done and what they still have left to do for each opportunity, lead, account, and contact.



Internal opportunity team prep call 10:00 AM | May 13
Start 5/13/2016 10:00 AM End 5/13/2016 11:00 AM

Review proposal with the team 11:30 AM | Tomorrow
Start 5/12/2016 11:30 AM End 5/12/2016 12:00 PM

A storm is coming... 10:04 AM | Today
Hey Ben! Acme wants to talk to you about incorporating their widgets into your kite design. Cheers, Gabrielle
To Address bfranklin@goflyakite.com From Address greynolds@goflyakite.com
Related To Acme - 1,200 Widgets

Meet with CIO Apr 28

Activity Timeline Considerations

When working with the activity timeline, keep the following in mind.

The configuration of page layouts and record types affects the tabs in the activity composer

Don't see the tabs for calls, tasks, events, or emails in the activity composer? Adjust your page layouts, record types, and user permissions. See Configure the Call, Task, and Event Tabs in the Activity Composer in Lightning Experience and Configure the Email Action in the Activity Composer in Lightning Experience.

The activity timeline replaces the Open Activities and Activity History related lists

On the detail page for objects that support activities, Lightning Experience doesn't display Open Activities or Activity History along with other related lists. It displays the activity timeline instead. Objects that support activities include opportunities, leads, accounts, contacts, and any custom objects on which you enable activities. You can customize the activity timeline using the Lightning App Builder.

You can customize the display and order of fields in the activity timeline

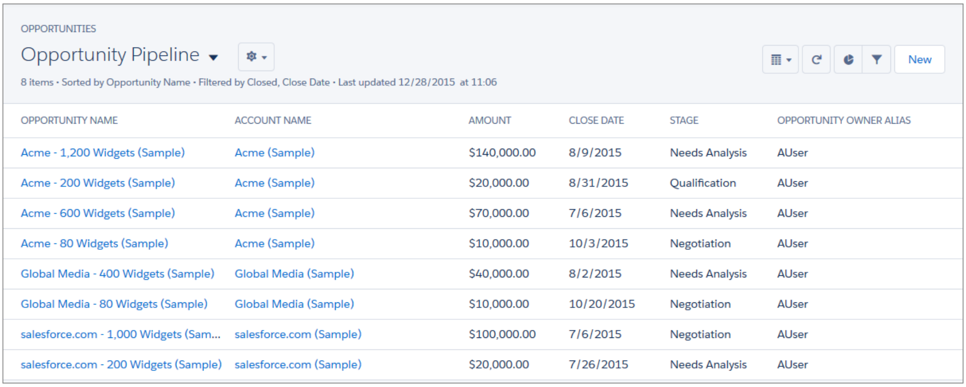
In the activity timeline, you can customize the display and order of fields for events, tasks, and logged calls using event and task compact layouts. However, even if you remove certain fields from a layout, they remain in the timeline because they contain essential activity information. For example, suppose that you remove the due date, the date and time, or the task status fields from the compact layouts. The event start date and time, the task checkbox, the task due date, and the call logging date still appear on activities in the timeline. The description field on events and the comments field on tasks also always appear in the timeline, although they aren't available in the compact layout. The remaining fields visible in the timeline reflect the fields you include in the compact layout.

The activity timeline icons aren't customizable

The icons for activity types (events, tasks, calls, and email) in the timeline aren't customizable.

Opportunity Home

The Opportunity home page in Lightning Experience looks a lot like other object home pages at first glance. You might not notice it at first, but an awesome feature is waiting to be discovered there.



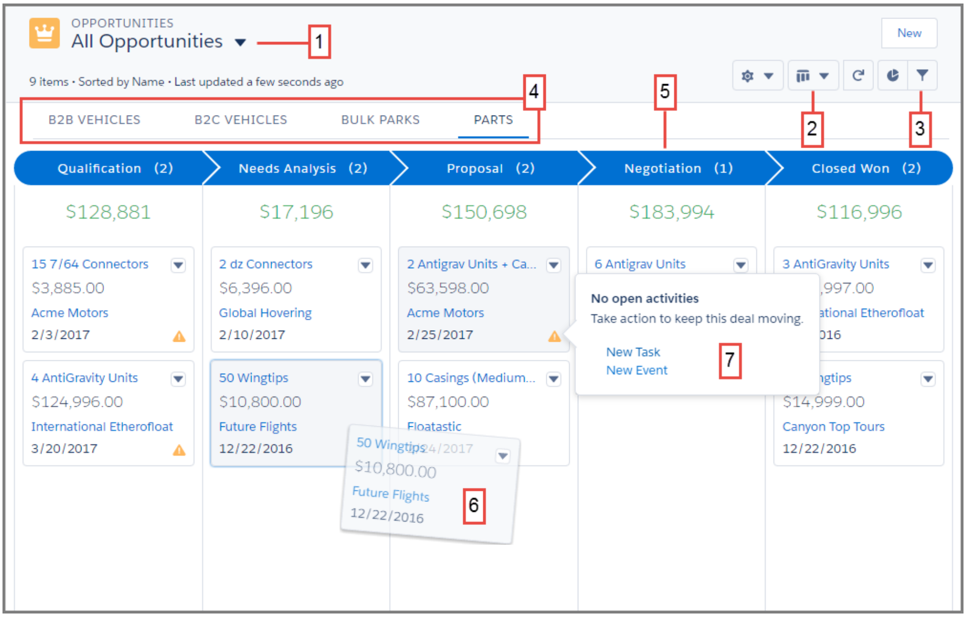
8 items • Sorted by Opportunity Name • Filtered by Closed, Close Date • Last updated 12/28/2015 at 11:06

OPPORTUNITY NAME	ACCOUNT NAME	AMOUNT	CLOSE DATE	STAGE	OPPORTUNITY OWNER ALIAS
Acme - 1,200 Widgets (Sample)	Acme (Sample)	$140,000.00	8/9/2015	Needs Analysis	AUser
Acme - 200 Widgets (Sample)	Acme (Sample)	$20,000.00	8/31/2015	Qualification	AUser
Acme - 600 Widgets (Sample)	Acme (Sample)	$70,000.00	7/6/2015	Needs Analysis	AUser
Acme - 80 Widgets (Sample)	Acme (Sample)	$10,000.00	10/3/2015	Negotiation	AUser
Global Media - 400 Widgets (Sample)	Global Media (Sample)	$40,000.00	8/2/2015	Needs Analysis	AUser
Global Media - 80 Widgets (Sample)	Global Media (Sample)	$10,000.00	10/20/2015	Negotiation	AUser
salesforce.com - 1,000 Widgets (Sam...	salesforce.com (Sample)	$100,000.00	7/6/2015	Negotiation	AUser
salesforce.com - 200 Widgets (Sample)	salesforce.com (Sample)	$20,000.00	7/26/2015	Needs Analysis	AUser

Kanban View

The Kanban view is a visual representation of all a sales rep's records for an object, such as Opportunities. The Kanban view isn't available on a few objects, such as knowledge and tasks.

The Kanban view for Opportunities is a visual representation of all of a sales rep's deals, organized by each stage in the pipeline. You can get to the Kanban view by selecting **Kanban** from the Displays menu on all list views except Recently Viewed.



OPPORTUNITIES
All Opportunities
New
9 items • Sorted by Name • Last updated a few seconds ago

B2B VEHICLES | B2C VEHICLES | BULK PARKS | PARTS

Qualification (2) — $128,881
- 15 7/64 Connectors, $3,885.00, Acme Motors, 2/3/2017
- 4 AntiGravity Units, $124,996.00, International Etherofloat, 3/20/2017

Needs Analysis (2) — $17,196
- 2 dz Connectors, $6,396.00, Global Hovering, 2/10/2017
- 50 Wingtips, $10,800.00, Future Flights, 12/22/2016

Proposal (2) — $150,698
- 2 Antigrav Units + Ca..., $63,598.00, Acme Motors, 2/25/2017
- 10 Casings (Medium..., $87,100.00, Floatastic

Negotiation (1) — $183,994
- 6 Antigrav Units

Closed Won (2) — $116,996
- 3 AntiGravity Units, ...997.00, ...ational Etherofloat, ...016
- ...ngtips, $14,999.00, Canyon Top Tours, 12/22/2016

50 Wingtips, $10,800.00, Future Flights, 12/22/2016

No open activities
Take action to keep this deal moving.
New Task
New Event

The records in the Kanban view are based on the selected list view (1). Reps can't view the Kanban for Recently Viewed list views. Easily toggle between the list view grid view and the Kanban view (2). Filter records to select a single record type or view a particular subset of records (3). Records are separated based on record type (4). Records are grouped into columns (5) based on the grouping field selected. Quickly move a record to a different stage by dragging the card (6). For opportunities, alerts tell how to keep a deal on track, for example, create a task or event (7).

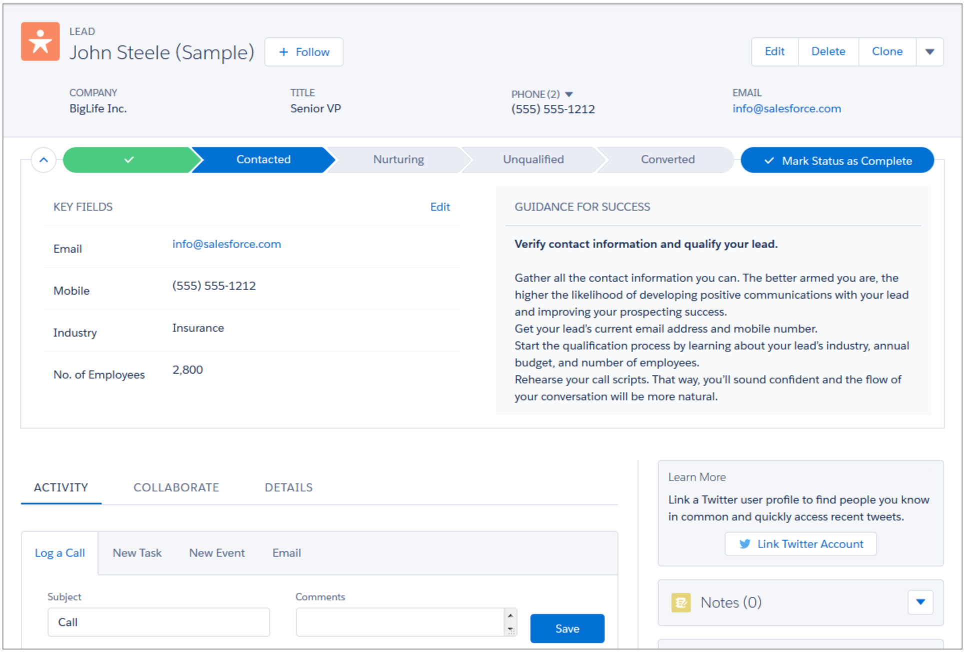
Here, your users can manage their opportunities through all phases of the pipeline, dragging and dropping opportunities from one column to another. A yellow triangle on an opportunity card can indicate three types of alerts: overdue tasks, no open activities, or no activity for 30 days. Users can click the triangle to create tasks and events right from the card. Items on the board vary based on which list view is open.

Tip: The Kanban view isn't just for opportunities--you can also access a Kanban view for almost any object you're viewing. .

Explore the Lead Workspace

Qualifying and converting leads is easy for your sales reps. Lightning Experience includes a lead workspace—command central where your reps track, update, and convert leads to contacts.

Just like with opportunities, the workspace for leads includes Sales Path. You set up sales paths to include specific fields and guidance for success in each stage of the lead qualification process.



LEAD
John Steele (Sample)

COMPANY
BigLife Inc.

TITLE
Senior VP

PHONE (2)
(555) 555-1212

EMAIL
info@salesforce.com

KEY FIELDS

Email	info@salesforce.com
Mobile	(555) 555-1212
Industry	Insurance
No. of Employees	2,800

GUIDANCE FOR SUCCESS

Verify contact information and qualify your lead.

Gather all the contact information you can. The better armed you are, the higher the likelihood of developing positive communications with your lead and improving your prospecting success.
Get your lead's current email address and mobile number.
Start the qualification process by learning about your lead's industry, annual budget, and number of employees.
Rehearse your call scripts. That way, you'll sound confident and the flow of your conversation will be more natural.

Learn More
Link a Twitter user profile to find people you know in common and quickly access recent tweets.

Subject
Call

Comments

To convert a lead, your reps click the **Converted** stage in the sales path. Then, they either select an account or create one on the spot. Reps can also create an opportunity.

CHAPTER 7 Explore More of Lightning Experience

In this chapter ...

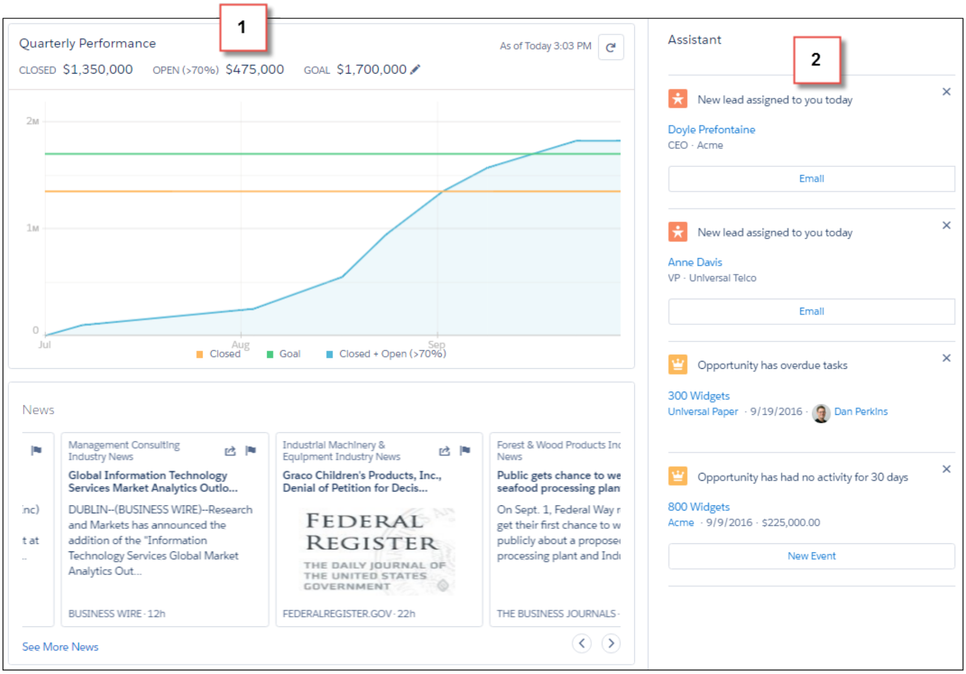
- The Home Page
- Object Home and List Views
- Accounts, Contacts, and Other Objects
- User Profile
- Tasks, Calendar, and Events
- Chatter in Lightning Experience

Opportunities and leads aren't the only places we've made improvements. There are slick new features to discover elsewhere, like the Home page, calendar, enhanced list views, and revamped home pages for objects like acccounts, contacts, and cases.

The Home Page

The Home page displays key items for each user's day. From the Home page, your users can manage their day, including viewing their quarterly performance summary and the most relevant tasks and updates. You can also use the Lightning App Builder to create custom Home pages that appear for different profiles.

Give your users access to opportunity details so that they can get the most out of the Home page.



Quarterly Performance

CLOSED $1,350,000 OPEN (>70%) $475,000 GOAL $1,700,000

As of Today 3:03 PM

News

Management Consulting Industry News

Global Information Technology Services Market Analytics Outlo...

DUBLIN--(BUSINESS WIRE)--Research and Markets has announced the addition of the "Information Technology Services Global Market Analytics Out...

BUSINESS WIRE · 12h

Industrial Machinery & Equipment Industry News

Graco Children's Products, Inc., Denial of Petition for Decis...

FEDERAL REGISTER THE DAILY JOURNAL OF THE UNITED STATES GOVERNMENT

FEDERALREGISTER.GOV · 22h

Forest & Wood Products Inc News

Public gets chance to we seafood processing plan

On Sept. 1, Federal Way r get their first chance to w publicly about a propose processing plant and Indu

THE BUSINESS JOURNALS ·

See More News

Assistant

New lead assigned to you today

Doyle Prefontaine
CEO · Acme

Email

New lead assigned to you today

Anne Davis
VP · Universal Telco

Email

Opportunity has overdue tasks

300 Widgets
Universal Paper · 9/19/2016 · Dan Perkins

Opportunity has had no activity for 30 days

800 Widgets
Acme · 9/9/2016 · $225,000.00

New Event

Performance Chart (1)

The performance chart displays data based on opportunities belonging to the user or the user's sales team. Only opportunities for the current sales quarter that are closed or open with a probability over 70% are displayed. Multicurrency is supported in the performance chart.

Curious about the numbers at the top of the chart? Here's what they mean.

- Closed—The sum of a user's closed opportunities.

- Open (>70%)—The sum of a user’s open opportunities with a probability over 70%. The blue line in the chart is the combined total of the closed opportunities and open opportunities with a probability over 70%.
- Goal—A user’s customizable sales goal for the quarter. This field is specific to the performance chart and has no impact on forecast quotas or any other type of goals. Click ✎ to set the goal.

Assistant (2)

The Assistant shows your users things they need to address, including new leads and activities related to opportunities.

Items in the Assistant appear in the following order:

- Leads assigned to you today
- Opportunities with overdue tasks
- Opportunities with no activity in 30 days
- Opportunities with no open activity

If your users don’t have access to activities on opportunities or if the opportunity pipeline is off, they instead see opportunities that have close dates over the next 90 days.

Note: In Salesforce Classic, the Home page has a Chatter feed. In Lightning Experience, that feed isn't there. Instead, Chatter (the "Feed") is on the navigation bar. When you want to access Chatter on a record, go to the Chatter tab.

The Assistant doesn’t show tasks due today or overdue tasks that aren’t tied to an opportunity. The Today’s Tasks component is an alternative that’s available on the Home page, and it shows a list of your tasks due today.

To populate the performance chart, Top Deals, and the Assistant, users must have:

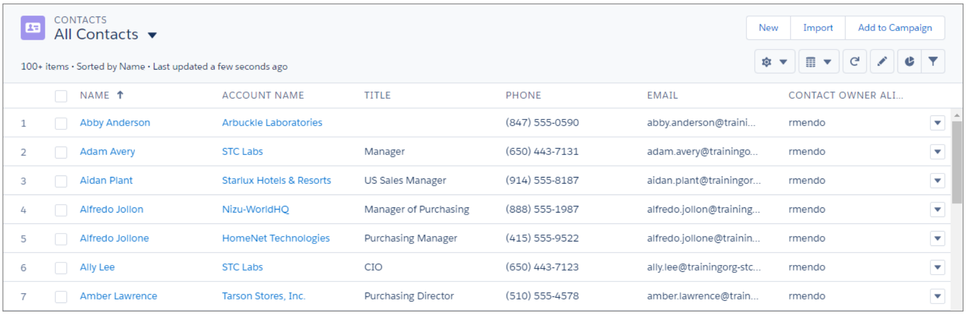
Table 3: Required Permissions for Home Features

Permission or Setting	Performance Chart	Top Deals	Assistant
Read access to the Opportunity object and sharing access to relevant opportunities	✓	✓	✓
Read access to the Opportunity object’s Amount field	✓	✓	
Read access to the Opportunity object’s Probability field	✓		

Permission or Setting	Performance Chart	Top Deals	Assistant
"Run Reports" user permission enabled for users	✓		
Closed opportunities or open opportunities with a probability over 70% during the current fiscal quarter	✓		
Read access to the Lead object			✓

Object Home and List Views

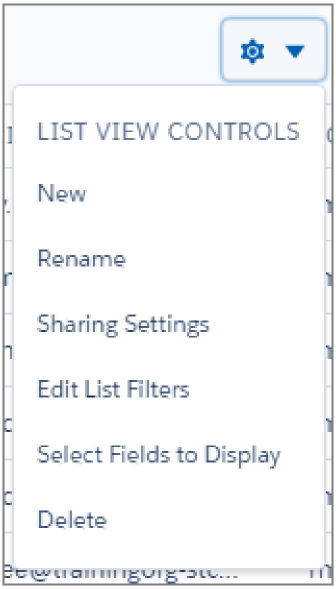
Salesforce Classic has a separate page for an object's home and another for its list views. In Lightning Experience, we've combined them!



100+ items • Sorted by Name • Last updated a few seconds ago

		NAME ↑	ACCOUNT NAME	TITLE	PHONE	EMAIL	CONTACT OWNER ALI...
1	□	Abby Anderson	Arbuckle Laboratories		(847) 555-0590	abby.anderson@traini...	rmendo
2	□	Adam Avery	STC Labs	Manager	(650) 443-7131	adam.avery@trainingo...	rmendo
3	□	Aidan Plant	Starlux Hotels & Resorts	US Sales Manager	(914) 555-8187	aidan.plant@trainingor...	rmendo
4	□	Alfredo Jollon	Nizu-WorldHQ	Manager of Purchasing	(888) 555-1987	alfredo.jollon@training...	rmendo
5	□	Alfredo Jollone	HomeNet Technologies	Purchasing Manager	(415) 555-9522	alfredo.jollone@trainin...	rmendo
6	□	Ally Lee	STC Labs	CIO	(650) 443-7123	ally.lee@trainingorg-stc...	rmendo
7	□	Amber Lawrence	Tarson Stores, Inc.	Purchasing Director	(510) 555-4578	amber.lawrence@train...	rmendo

What's that gear menu for? It contains options for managing your list views.



In the charts panel (◔), you can change the type of chart or create a new one. Click ▼ to add, set, or remove filters.

The name of the recent records list that displays by default on the home page for standard and custom objects is different in Lightning Experience. It's called Recently Viewed. There's another list view available for most standard objects that has the object type specified in the name. For example, the list view for accounts is called Recently Viewed Accounts. These two list views, Recently Viewed and Recently Viewed *`Object`*, show the same records. Neither list is deletable, filterable, or editable.

As an administrator, you can configure an object's Recently Viewed search layout for your users. The search layout controls what all users see when they land on that object's home page. In Lightning Experience, from Setup, find the object in the Object Manager, then scroll to its Search Layout related list and edit the Search Results search layout.

You can also go directly to an object's details page by selecting **Edit Object** from the Setup menu.

Limitations and Differences

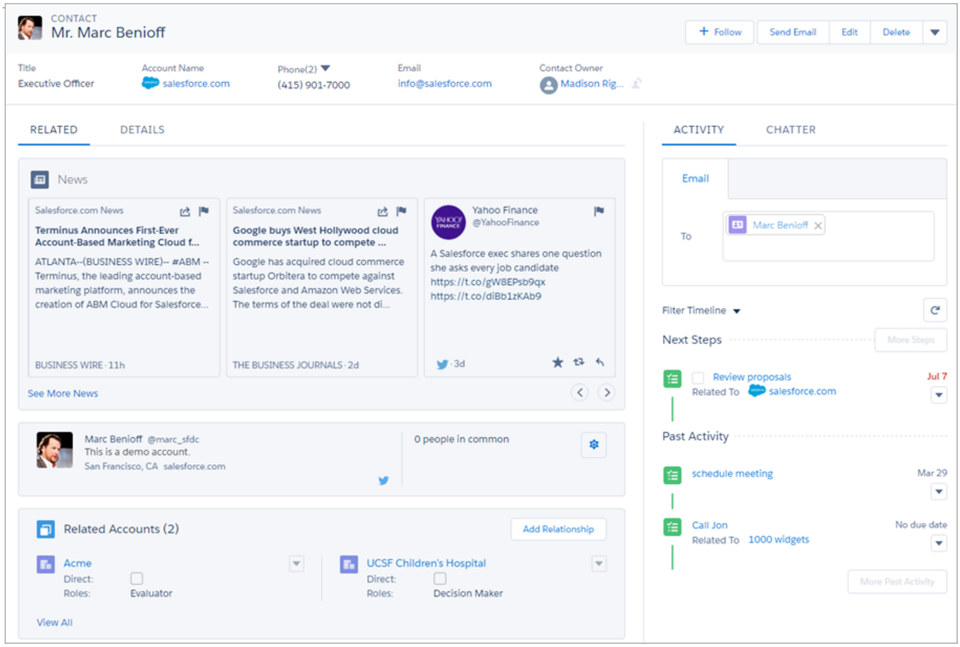
Some features and links that are available on object home pages in Salesforce Classic aren't available on comparable pages in Lightning Experience. You can still view, create, edit, and delete list views on these pages, but the steps you take for these tasks are different, and there are differences in how list views display and behave. Here are a few examples. The rest can be found in Lightning Experience Considerations.

- Navigation through list views has changed. Because list views have infinite scrolling, you can't specify the number of records to show per page. Also, you can't navigate through pages with arrow buttons, and you can't jump to a specific page of results.
- The list views drop-down menu shows up to 2,000 views. The menu loads views in batches of 50 as you scroll down. Recently viewed lists appear first, followed by other list views in alphabetical order. Recently viewed lists and the Search box appear only if there are 11 or more list views.
- Instead of an alphabet rolodex for list views, the List View Controls drop-down menu has auto-complete search that's enabled whenever there are more than 10 list views.
- You can change the columns that appear and their order by choosing Select Fields to Display from the List View Controls drop-down menu.
- Your users can reorder the columns of the Recently Viewed *`Object`* list in Salesforce Classic, and the changes are reflected in Lightning Experience.
- Custom buttons and custom actions aren't supported for list views or list view items.

Accounts, Contacts, and Other Objects

Opportunities and leads have special workspaces. The other objects have a different structure. Some elements, like related lists and the Activity and Chatter tabs, appear in different places than we saw on opportunities and leads.

For example, here's a contact page.



Mr. Marc Benioff

Title: Executive Officer
Account Name: salesforce.com
Phone(2): (415) 901-7000
Email: info@salesforce.com
Contact Owner: Madison Rig...

Salesforce.com News
Terminus Announces First-Ever Account-Based Marketing Cloud f...
ATLANTA--(BUSINESS WIRE)-- #ABM -- Terminus, the leading account-based marketing platform, announces the creation of ABM Cloud for Salesforce...
BUSINESS WIRE · 11h

Salesforce.com News
Google buys West Hollywood cloud commerce startup to compete ...
Google has acquired cloud commerce startup Orbitera to compete against Salesforce and Amazon Web Services. The terms of the deal were not di...
THE BUSINESS JOURNALS · 2d

Yahoo Finance @YahooFinance
A Salesforce exec shares one question she asks every job candidate https://t.co/gW8EPsb9qx https://t.co/dIBb1zKAb9
3d

Marc Benioff @marc_sfdc
This is a demo account.
San Francisco, CA salesforce.com
0 people in common

Acme
Direct: □
Roles: Evaluator

UCSF Children's Hospital
Direct: □
Roles: Decision Maker

To: Marc Benioff

□ Review proposals — Jul 7
Related To salesforce.com

schedule meeting — Mar 29

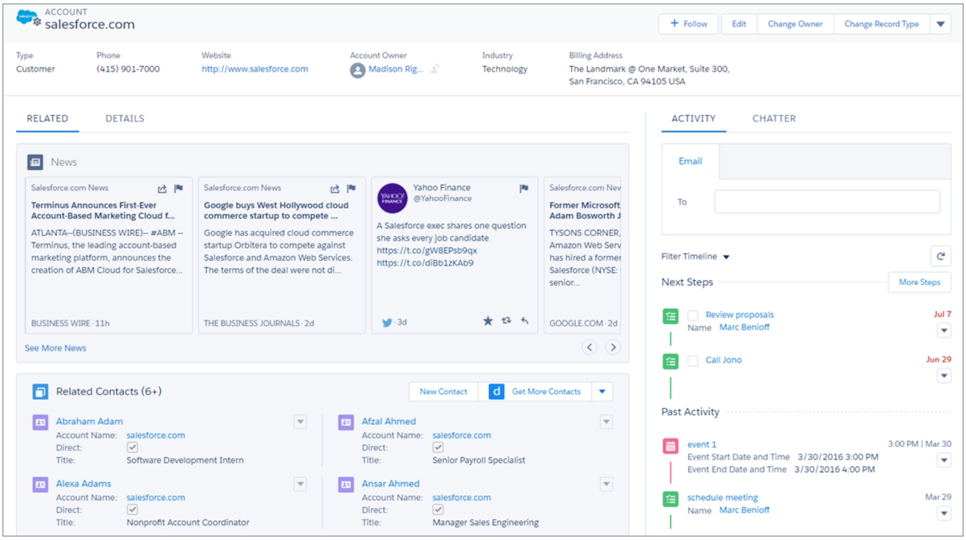
Call Jon — No due date
Related To 1000 widgets

The highlights panel is in the same place across all objects. But here there's no Sales Path, and the Activity and Chatter tabs are together in the right hand column. Related lists are in their own tab in the main part of the page, alongside the Details tab.

Some objects have special components that appear on their pages (for example, accounts, contacts, leads, and opportunities include News), but the overall structure remains the same across all objects.

Accounts and Contacts

The Contacts to Multiple Accounts feature lets you relate a single contact to multiple accounts so that you can easily track the relationships between people and businesses—without creating duplicate records. When the feature is set up, account records include the Related Contacts related list and contact records include the Related Accounts related list.



ACCOUNT
salesforce.com

Type: Customer
Phone: (415) 901-7000
Website: http://www.salesforce.com
Account Owner: Madison Rig...
Industry: Technology
Billing Address: The Landmark @ One Market, Suite 300, San Francisco, CA 94105 USA

News

Salesforce.com News
Terminus Announces First-Ever Account-Based Marketing Cloud f...
ATLANTA--(BUSINESS WIRE)-- #ABM -- Terminus, the leading account-based marketing platform, announces the creation of ABM Cloud for Salesforce...
BUSINESS WIRE · 11h

Salesforce.com News
Google buys West Hollywood cloud commerce startup to compete ...
Google has acquired cloud commerce startup Orbitera to compete against Salesforce and Amazon Web Services. The terms of the deal were not di...
THE BUSINESS JOURNALS · 2d

Yahoo Finance
@YahooFinance
A Salesforce exec shares one question she asks every job candidate https://t.co/gW8EPsb9qx https://t.co/diBb1zKAb9
· 3d

Salesforce.com Nev
Former Microsoft Adam Bosworth J
TYSONS CORNER, Amazon Web Serv has hired a former Salesforce (NYSE: senior...
GOOGLE.COM · 2d

Related Contacts (6+)

Abraham Adam
Account Name: salesforce.com
Direct: ☑
Title: Software Development Intern

Afzal Ahmed
Account Name: salesforce.com
Direct: ☑
Title: Senior Payroll Specialist

Alexa Adams
Account Name: salesforce.com
Direct: ☑
Title: Nonprofit Account Coordinator

Ansar Ahmed
Account Name: salesforce.com
Direct: ☑
Title: Manager Sales Engineering

Next Steps

□ Review proposals — Jul 7
Name: Marc Benioff

□ Call Jono — Jun 29

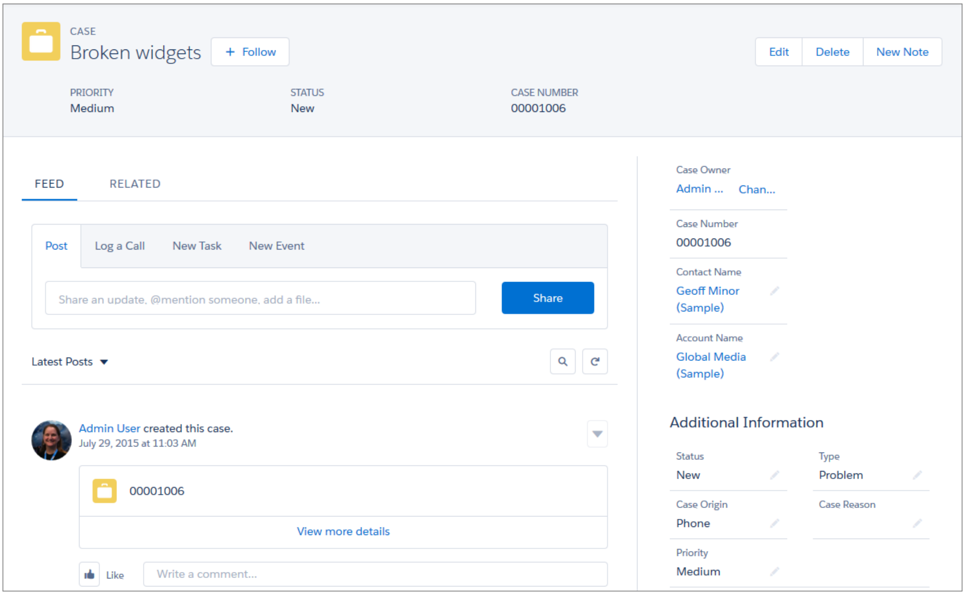
Past Activity

event 1 — 3:00 PM | Mar 30
Event Start Date and Time 3/30/2016 3:00 PM
Event End Date and Time 3/30/2016 4:00 PM

schedule meeting — Mar 29
Name: Marc Benioff

Cases

Cases look a little bit different than other records. Cases display a feed first rather than record details or related information, because support agents mainly collaborate and work with activities in a feed. Details appear next to the feed. You don't see the Activity or Collaboration tabs because this information displays directly in the feed.



CASE
Broken widgets

PRIORITY
Medium

STATUS
New

CASE NUMBER
00001006

Share an update, @mention someone, add a file...

Admin User created this case.
July 29, 2015 at 11:03 AM

00001006

Write a comment...

Case Owner
Admin ... Chan...

Case Number
00001006

Contact Name
Geoff Minor (Sample)

Account Name
Global Media (Sample)

Additional Information

Status
New

Type
Problem

Case Origin
Phone

Case Reason

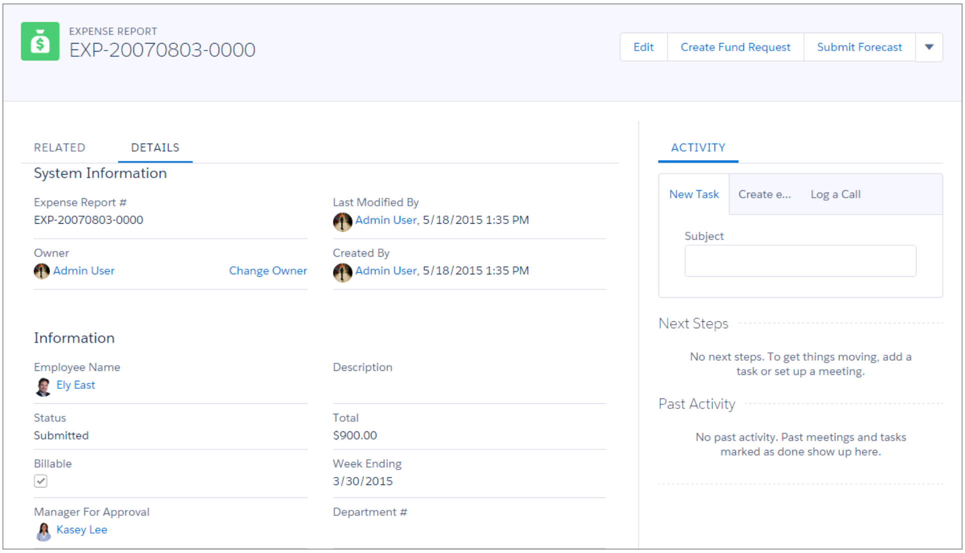
Priority
Medium

If you turn off Case Feed, cases look like other records. Activities and collaboration appear, but the feed-first design disappears and adds more clicks and context-switching to cases. You can use collaboration (Chatter) instead of feed, but collaboration doesn't include useful case feed publishers, such as email, log a call, or change status.

To fully benefit from the new feed-first design on cases, re-create the standard case feed publishers. The standard publishers aren't available, but you can quickly re-create them as quick actions on the Case object. One benefit of re-creating the publishers as quick actions is that they appear on mobile devices, whereas standard case feed publishers don't.

Custom Objects

Custom objects are supported in Lightning Experience, and custom object home pages contain the same standard elements as other objects, such as details, related lists, highlights panel, activities and a feed (Chatter).



EXPENSE REPORT
EXP-20070803-0000

System Information

Expense Report #
EXP-20070803-0000

Last Modified By
Admin User, 5/18/2015 1:35 PM

Owner
Admin User Change Owner

Created By
Admin User, 5/18/2015 1:35 PM

Information

Employee Name
Ely East

Description

Status
Submitted

Total
$900.00

Billable
☑

Week Ending
3/30/2015

Manager For Approval
Kasey Lee

Department #

Subject

Next Steps

No next steps. To get things moving, add a task or set up a meeting.

Past Activity

No past activity. Past meetings and tasks marked as done show up here.

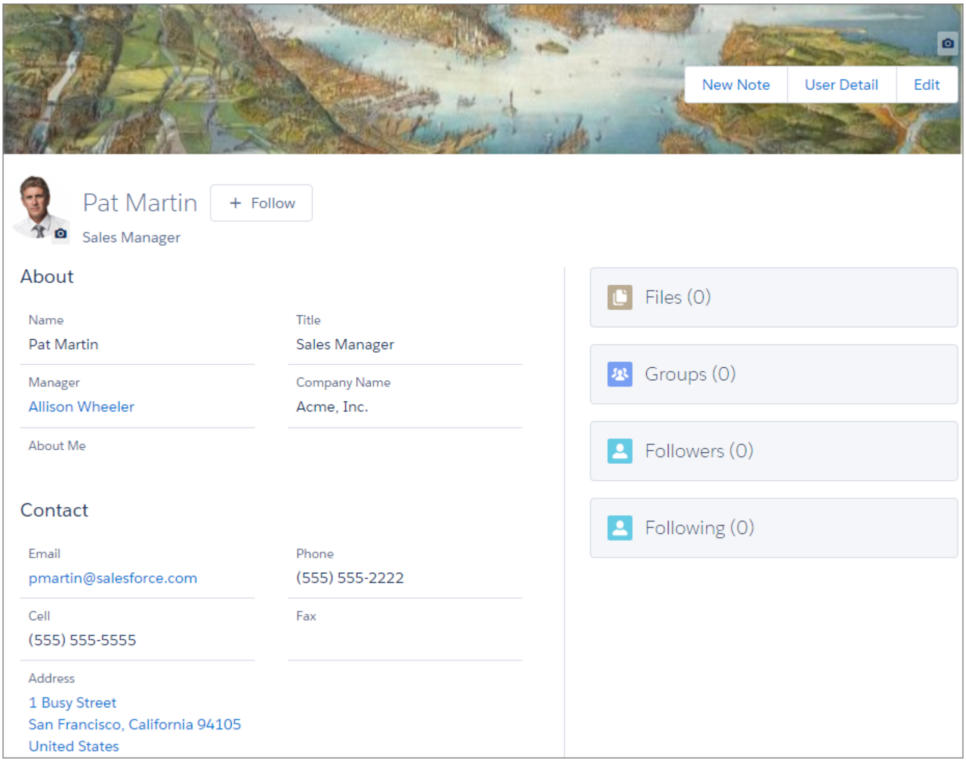
On this particular custom object, Expense Report, feed tracking hasn't been enabled. So, the Chatter tab, which you'd normally see next to Activity, doesn't appear.

Tip: On object home and record pages, you can select Edit Object from the Setup menu to go directly to that object's detail page.

User Profile

The user profile page has been streamlined in Lightning Experience. From here, you can follow a user, see their details, edit their information (if you have permission), and see team members, files, groups, and other related information.

As with other objects, the pertinent details are in the main part of the page and the related lists are in the right-hand column. One unique thing about the user profile page is that the Google map that you see by default on address fields for other objects is disabled for users.



Pat Martin

Sales Manager

About

Name: Pat Martin

Title: Sales Manager

Manager: Allison Wheeler

Company Name: Acme, Inc.

About Me

Contact

Email: pmartin@salesforce.com

Phone: (555) 555-2222

Cell: (555) 555-5555

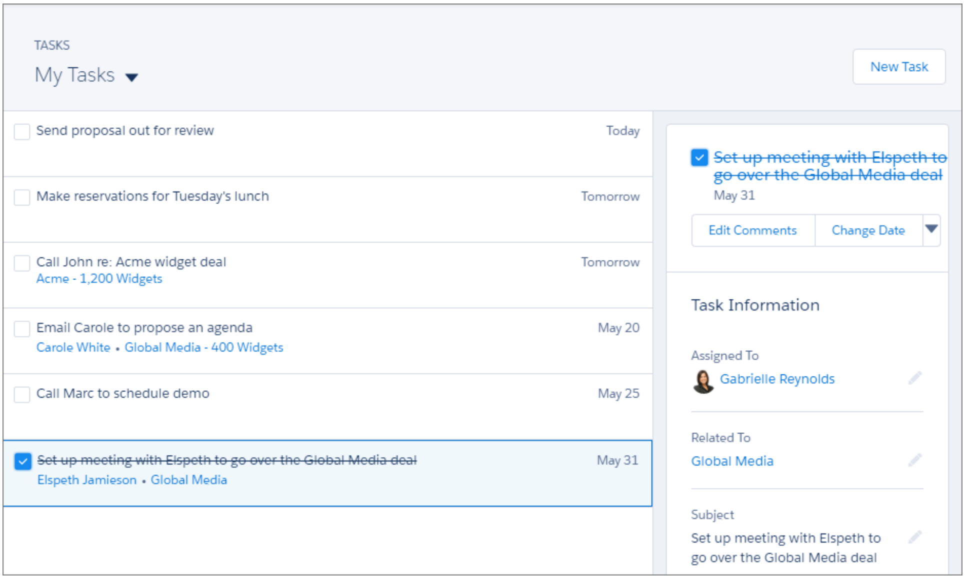
Fax

Address:
1 Busy Street
San Francisco, California 94105
United States

Tasks, Calendar, and Events

There's a better way to track the work you need to do to move deals ahead, with some nifty enhancements to tasks and calendars in Lightning Experience.

Tasks



TASKS

My Tasks ▾

New Task

□ Send proposal out for review — Today

□ Make reservations for Tuesday's lunch — Tomorrow

□ Call John re: Acme widget deal — Tomorrow
Acme - 1,200 Widgets

□ Email Carole to propose an agenda — May 20
Carole White • Global Media - 400 Widgets

□ Call Marc to schedule demo — May 25

☑ ~~Set up meeting with Elspeth to go over the Global Media deal~~ — May 31
Elspeth Jamieson • Global Media

☑ ~~Set up meeting with Elspeth to go over the Global Media deal~~
May 31

Edit Comments | Change Date | ▾

Task Information

Assigned To
Gabrielle Reynolds

Related To
Global Media

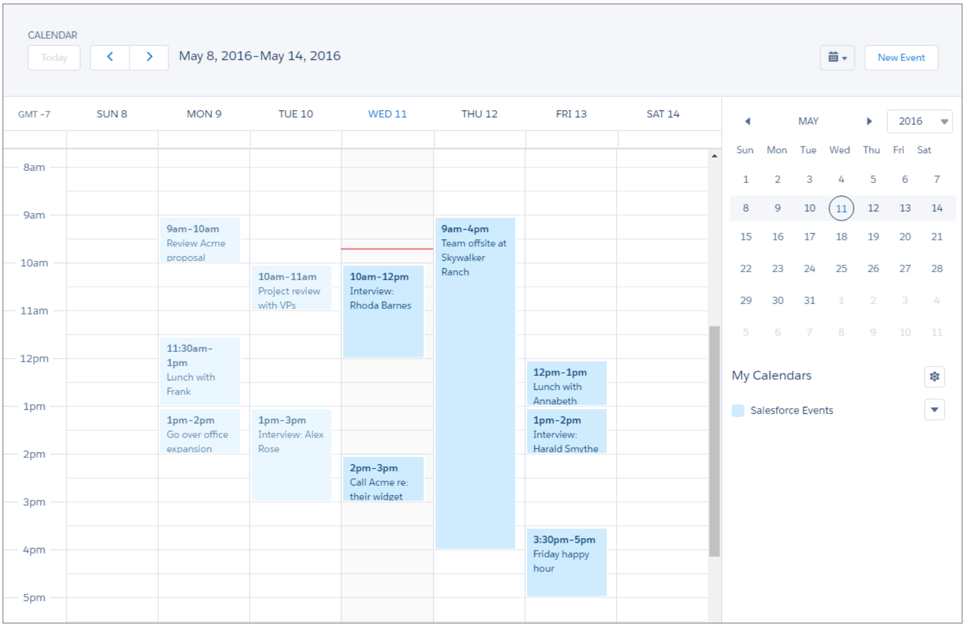
Subject
Set up meeting with Elspeth to go over the Global Media deal

- View a list of all your open tasks, showing the opportunities, accounts, and other records they're related to. See and edit details right there on the same page.
- Switch to views of tasks due today, completed tasks, and tasks you've delegated.
- If you have 10 or more overdue tasks, they're collapsed under **Show More Overdue Tasks**.

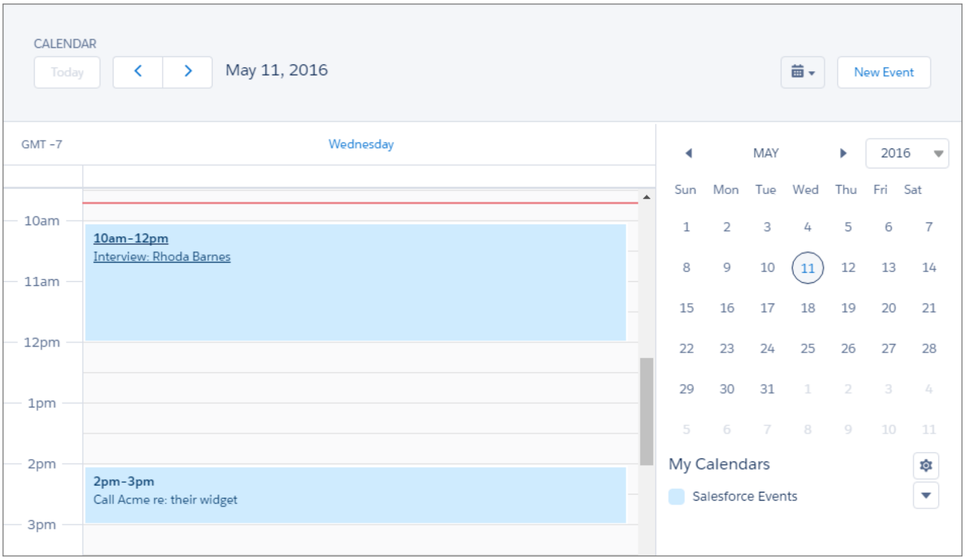
Use compact layouts to customize the display and order of fields for items in the task list. However, certain fields remain in the task list even if you remove them from a layout, because they contain essential task information. For example, suppose that you remove the status and due date fields from a compact layout. The task checkbox and the due date still appear on tasks in the list. The remaining fields visible in the list reflect the fields you include in the compact layout.

Calendar

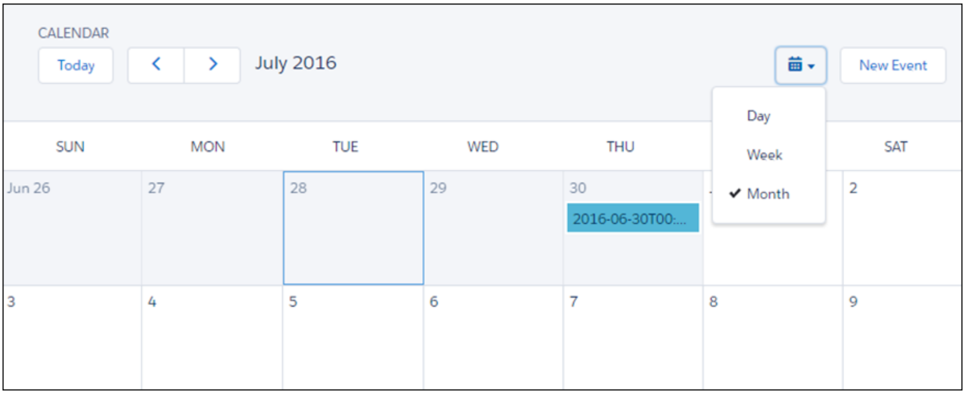
Calendar week view:


CALENDAR
Today
May 8, 2016–May 14, 2016
New Event
GMT -7
SUN 8
MON 9
TUE 10
WED 11
THU 12
FRI 13
SAT 14
8am
9am
10am
11am
12pm
1pm
2pm
3pm
4pm
5pm
9am–10am Review Acme proposal
10am–11am Project review with VPs
10am–12pm Interview: Rhoda Barnes
9am–4pm Team offsite at Skywalker Ranch
11:30am–1pm Lunch with Frank
12pm–1pm Lunch with Annabeth
1pm–2pm Go over office expansion
1pm–3pm Interview: Alex Rose
1pm–2pm Interview: Harald Smythe
2pm–3pm Call Acme re: their widget
3:30pm–5pm Friday happy hour
MAY
2016
Sun Mon Tue Wed Thu Fri Sat
1 2 3 4 5 6 7
8 9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31 1 2 3 4
5 6 7 8 9 10 11
My Calendars
Salesforce Events

Calendar day view:



Calendar month view:

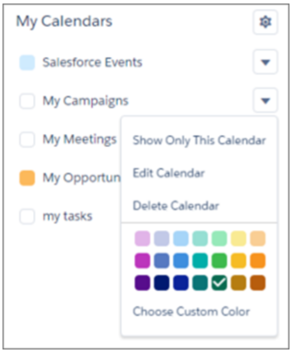


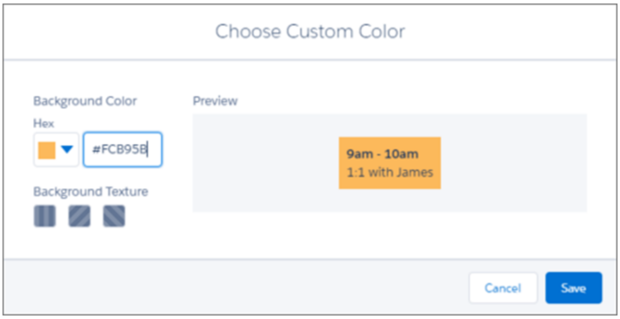
When you view, create, and edit events in Lightning Experience, the calender is streamlined and easier to use. The calendar displays all events owned by a user, including events outside a user's business hours, in the time zone selected in your Salesforce settings.

Let's say you want to track campaigns, cases, retail store events, opportunities, and more. Just add calendars for all of these things. Choose a field to track on any standard or custom object. The calendar displays data in that field as calendar items.

Users can customize most calendars by applying a list view, and they can edit and delete calendars they've created.

You can distinguish calendars by color and texture.





Hex #FCB95B

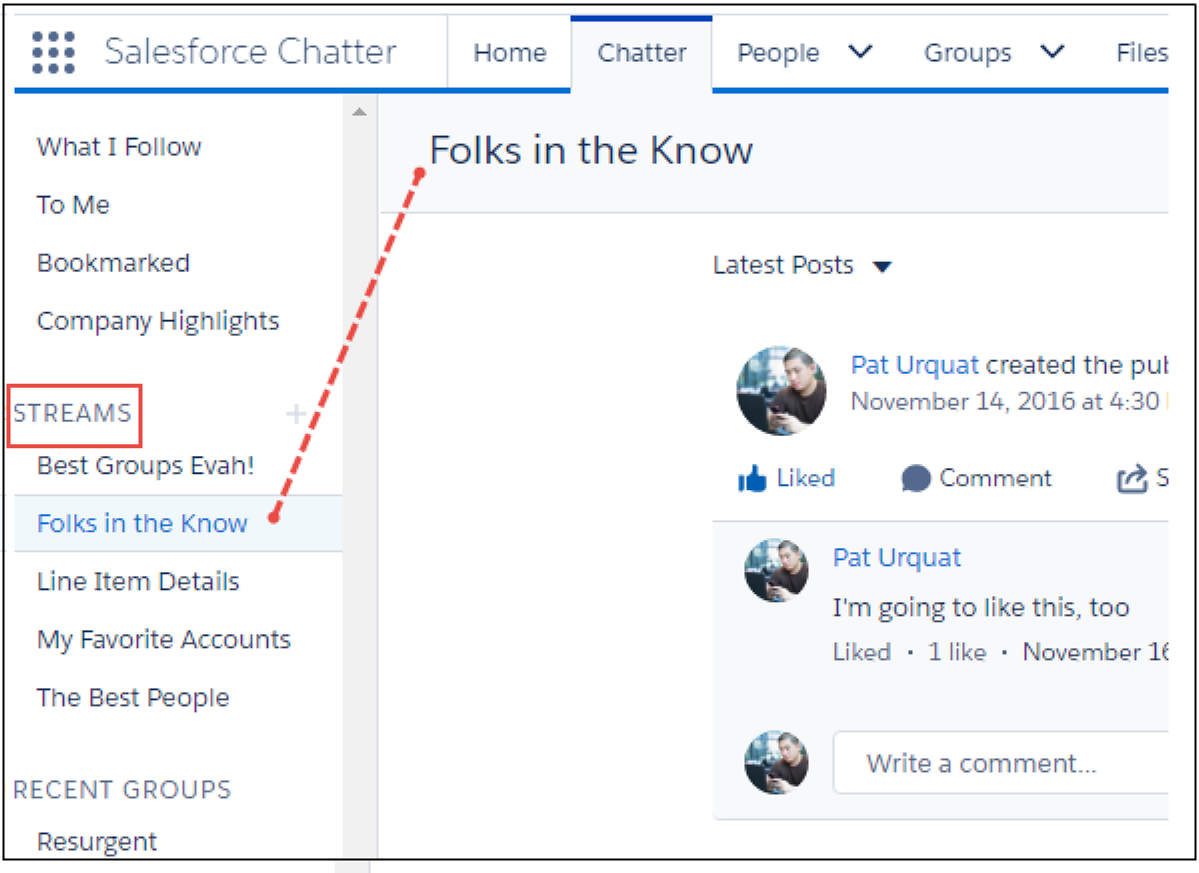
Calendar views display up to 150 items, including items from calendars you create.

You can’t set or view event reminders; create or edit recurring events; or create, edit, accept, or decline meeting invites. However, suppose that a Salesforce Classic user creates a series of recurring events. Lightning Experience users see the individual events on their calendars. Similarly, Lightning Experience users see individual events that they’ve been invited to by Salesforce Classic users.

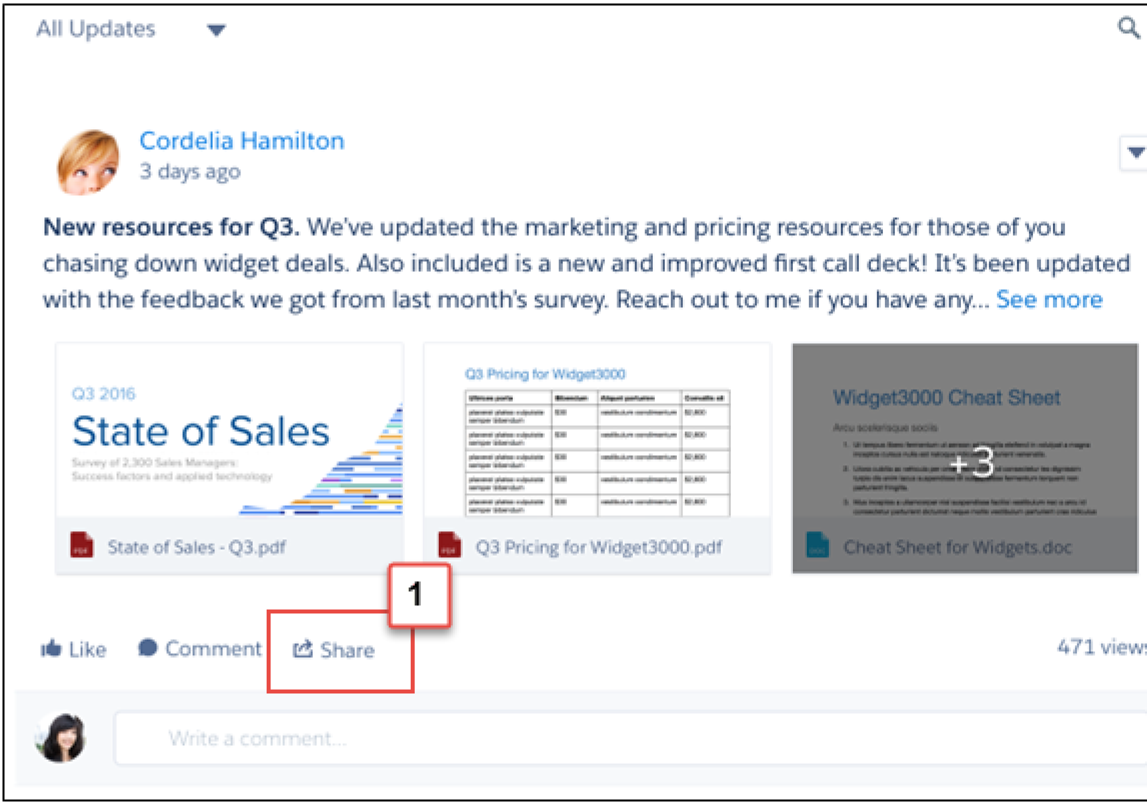
Chatter in Lightning Experience

Wondering what’s up with Chatter in Lightning Experience? Streams, for one. Combine up to 25 records into one feed—called a stream. Create up to five streams. Sharing’s come to Lightning: Now you can share a post with your followers. Enjoy increased relevance in your All Company feed, now named Company Highlights, brought to you by Einstein Artificial Intelligence (AI). We made posts and comments even easier with shortcut keys. And a typing indicator shows you when someone’s typing a comment.

Chatter streams are custom feeds that you create from the information that’s most useful to you. Look for streams on your Chatter home page. Create up to five streams that combine posts from up to 25 different feeds and feed types. Create streams that combine feeds from people, groups, and records, like accounts, opportunities, cases, and more.



When someone posts something you'd like more people to see, you can share it with your followers. If you like, you can add your own introductory remarks before you post. Click **Share** to get started (1).



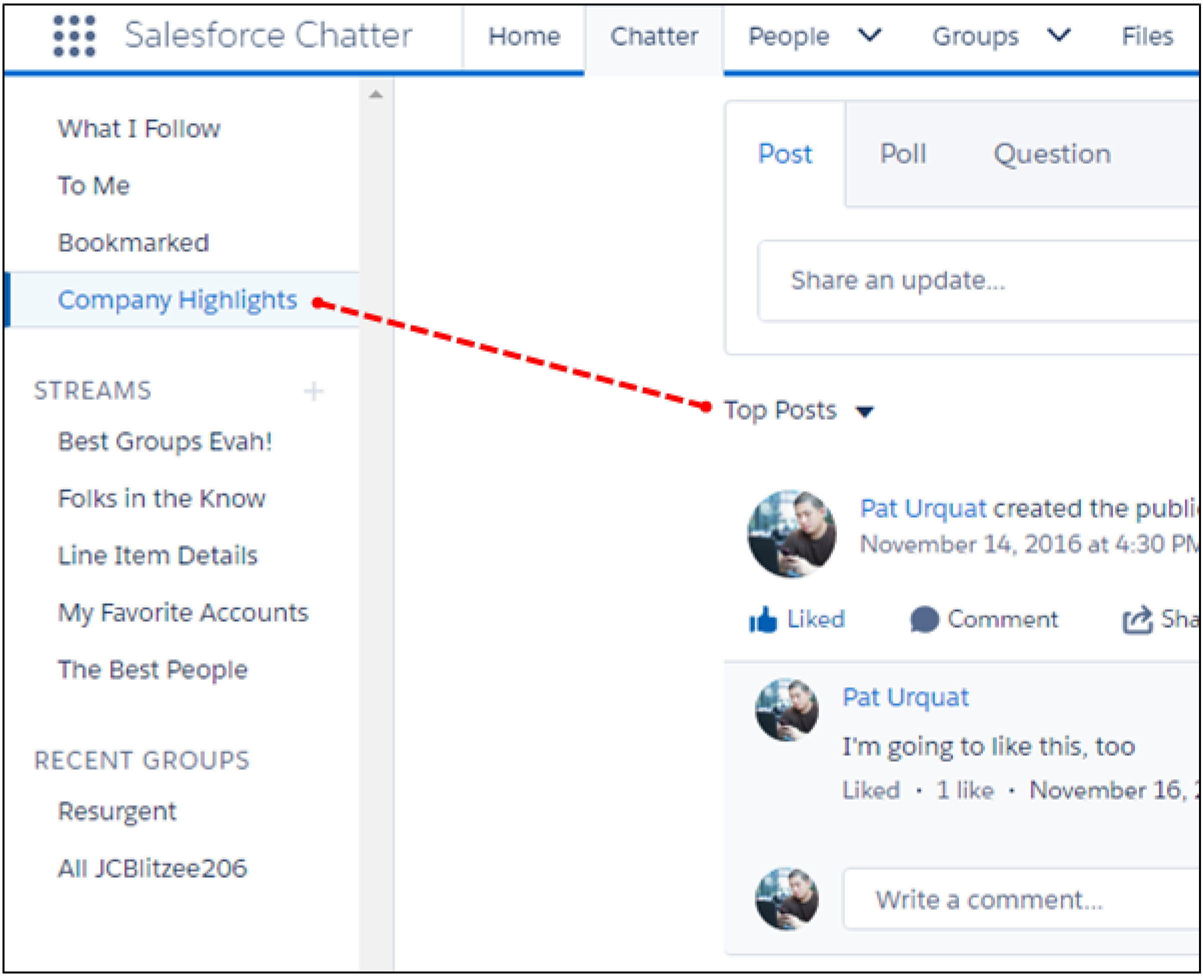
All Updates

Cordelia Hamilton
3 days ago

New resources for Q3. We've updated the marketing and pricing resources for those of you chasing down widget deals. Also included is a new and improved first call deck! It's been updated with the feedback we got from last month's survey. Reach out to me if you have any... See more

Q3 2016
State of Sales
Survey of 2,300 Sales Managers:
Success factors and applied technology
State of Sales - Q3.pdf
Q3 Pricing for Widget3000
Q3 Pricing for Widget3000.pdf
Widget3000 Cheat Sheet
Arcu scelerisque sociis
+3
Cheat Sheet for Widgets.doc
1
Like
Comment
Share
471 views
Write a comment...

Your All Company feed is now named Company Highlights. The Company Highlights feed incorporates Einstein AI to bring you a ranked view of what's trending in your org (Top Posts). Now you can easily catch up on the hottest news from across your company.



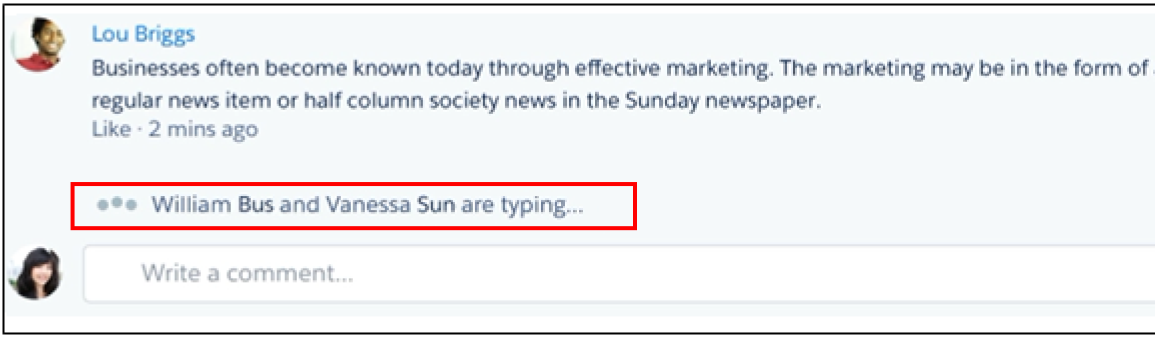
Pat Urquat created the publi
November 14, 2016 at 4:30 PM

Pat Urquat
I'm going to like this, too
Liked · 1 like · November 16, 2

Use shortcut keys to post and comment. On Windows, enter content and press Ctrl+Enter. On macOS, press either Control+Enter or Command+Enter.

Note: In questions, shortcut keys work with question details, but not on the question itself. So if you just ask a question, the shortcuts don't work. But if you include details with the question, shortcuts work.

We enhanced live comments for better usability. For example, an animation now shows you when people are commenting. Right by that animation is a list of up to three people who are actively typing comments.



Lou Briggs

Businesses often become known today through effective marketing. The marketing may be in the form of
regular news item or half column society news in the Sunday newspaper.

Like · 2 mins ago

William Bus and Vanessa Sun are typing...

Write a comment...

CHAPTER 8 Reports and Dashboards in Lightning Experience

In this chapter ...

- Report and Dashboard Home Pages
- Create Beautiful Dashboards Using a Feature-Rich Editor
- Present and Share Information in Interactive Dashboards
- Get More Information from Interactive Charts and Filters on the Report Run Page
- Reports and Dashboards: Compatibility Between Lightning Experience and Salesforce Classic

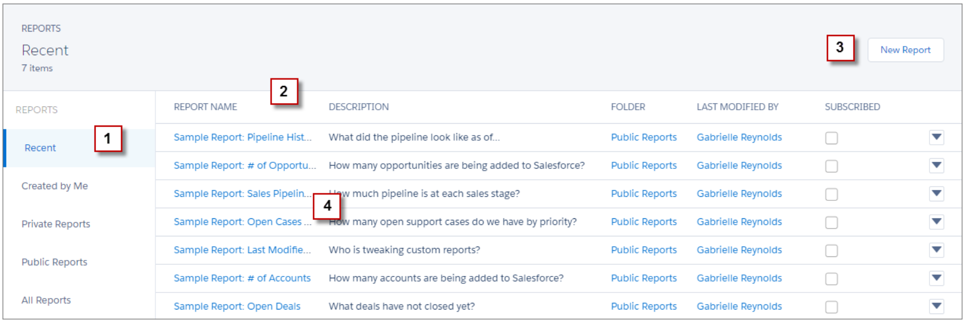
We've redesigned reports and dashboards for Lightning Experience, making them more interactive, easier to navigate, and easier to edit. Let's see what reports and dashboards in Lightning Experience can do!

Here's a quick overview:

- Get more information from interactive charts without needing to drill into reports (the charts look fabulous, too!)
- Create dashboards with more than three columns and column-spanning components using the lightning dashboard editor
- Edit filters while viewing a report without launching the report builder and see data update in real time
- Show or hide a chart, totals, and details from the report
- Find, manage, and create reports and dashboards more easily via redesigned home pages
- Get started with ready-to-go sales and service dashboards
- Work with your reports and dashboards from Salesforce Classic in Lightning Experience

Report and Dashboard Home Pages

We've redesigned the Reports and Dashboards home pages in Lightning Experience so that your users can find and create reports and dashboards more easily.



REPORTS

Recent

7 items

New Report

REPORTS

- Recent
- Created by Me
- Private Reports
- Public Reports
- All Reports

REPORT NAME	DESCRIPTION	FOLDER	LAST MODIFIED BY	SUBSCRIBED	
Sample Report: Pipeline Hist...	What did the pipeline look like as of...	Public Reports	Gabrielle Reynolds	□	▼
Sample Report: # of Opportu...	How many opportunities are being added to Salesforce?	Public Reports	Gabrielle Reynolds	□	▼
Sample Report: Sales Pipelin...	How much pipeline is at each sales stage?	Public Reports	Gabrielle Reynolds	□	▼
Sample Report: Open Cases ...	How many open support cases do we have by priority?	Public Reports	Gabrielle Reynolds	□	▼
Sample Report: Last Modifie...	Who is tweaking custom reports?	Public Reports	Gabrielle Reynolds	□	▼
Sample Report: # of Accounts	How many accounts are being added to Salesforce?	Public Reports	Gabrielle Reynolds	□	▼
Sample Report: Open Deals	What deals have not closed yet?	Public Reports	Gabrielle Reynolds	□	▼

Find a report or dashboard using filtered lists and folders (**1**). Filtered lists are the fastest way to find the report or dashboard you're looking for. For example, you can quickly find the Open Deals report you were reading last Friday in Recent. Folders let you group related reports or dashboards, so they're easy to find again later.

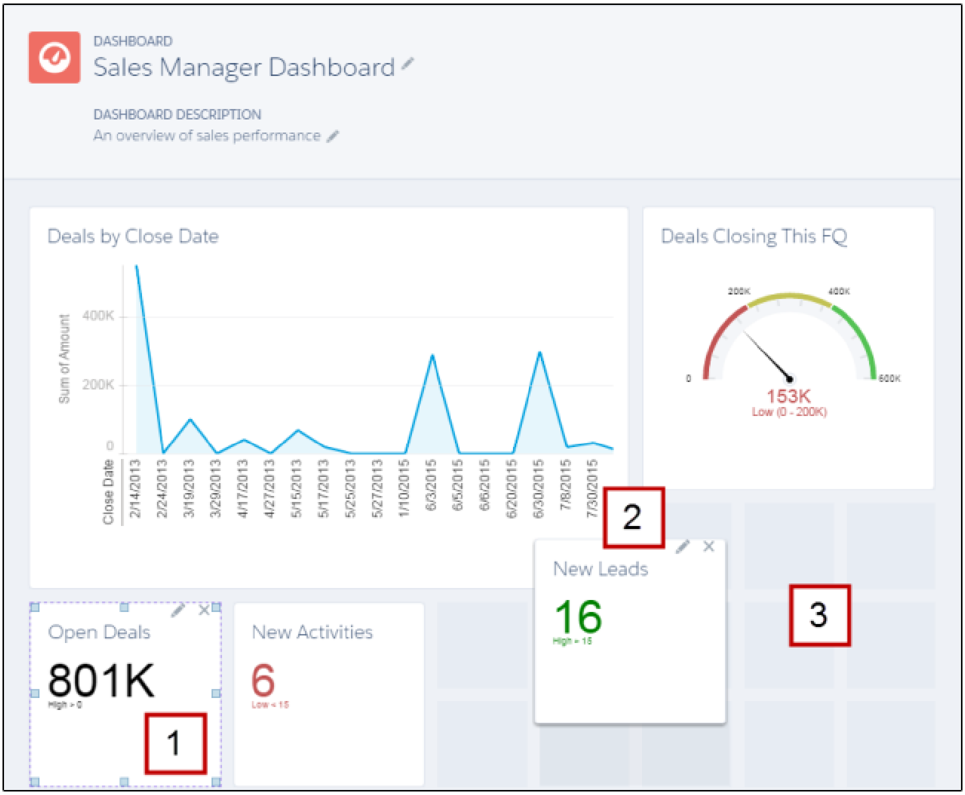
Click a column heading (**2**) to sort by name, folder, creator, or whoever last modified a report or dashboard. Click again to reverse the sort order.

You can build a new report or dashboard (**3**) right from this page. In Lightning Experience, you build reports just like you did in Salesforce Classic using the report builder. On the Dashboards home page, clicking **New Dashboard** launches the lightning dashboard editor, which we'll look at in more depth next.

To run and view a report or open a dashboard, click its name (**4**).

Create Beautiful Dashboards Using a Feature-Rich Editor

Lightning Experience introduces a new dashboard editor that'll usher in a whole new generation of dashboards.

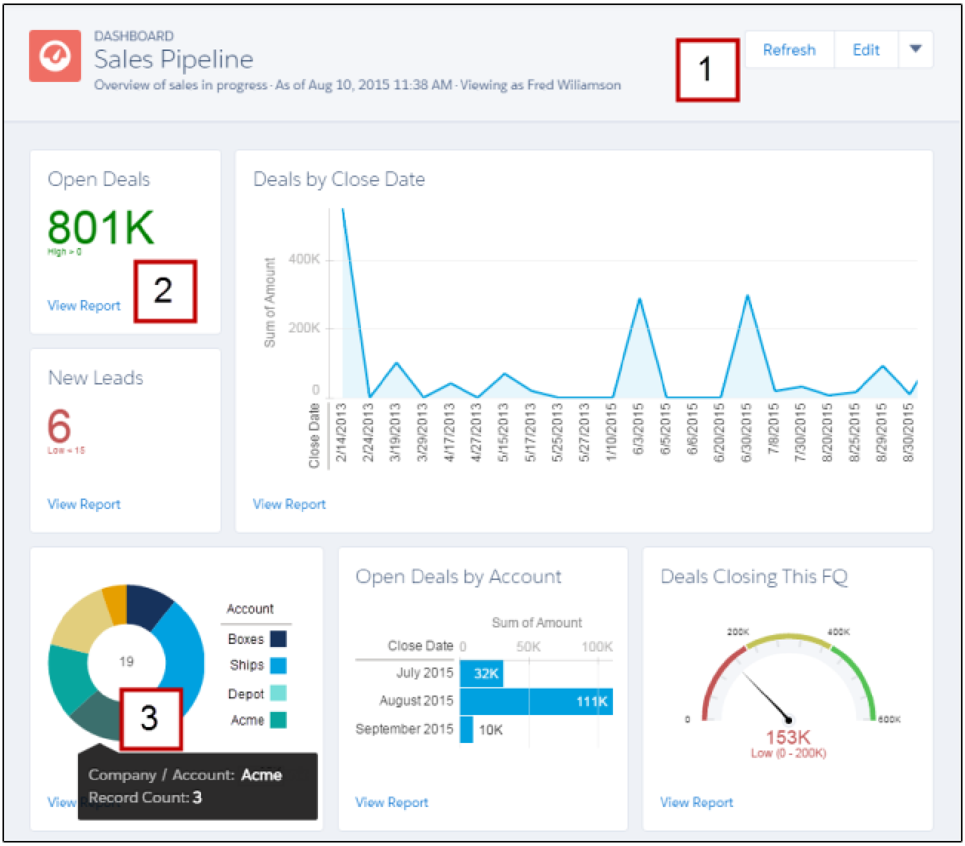


Drag the corners and sides of dashboard components to make them bigger or smaller (**1**). Components can span multiple columns and rows, so you can show more fields on a graph without needing to scroll. Charts automatically resize to match component size. Click ✏ to choose the report that provides data for the component, chart type, the report data drawn in the chart, chart title, or range.

Arrange dashboard components more easily than ever by dragging them anywhere on the flexible grid (**2**). Not only that, but you can create dashboards with more than three columns (**3**)! The foundation of each dashboard is a responsive grid, so you can compare metrics side-by-side-by-side.

Present and Share Information in Interactive Dashboards

Interactive dashboard components give viewers more information and link to data-supplying reports.



DASHBOARD

Sales Pipeline

Overview of sales in progress · As of Aug 10, 2015 11:38 AM · Viewing as Fred Williamson

Company / Account: Acme

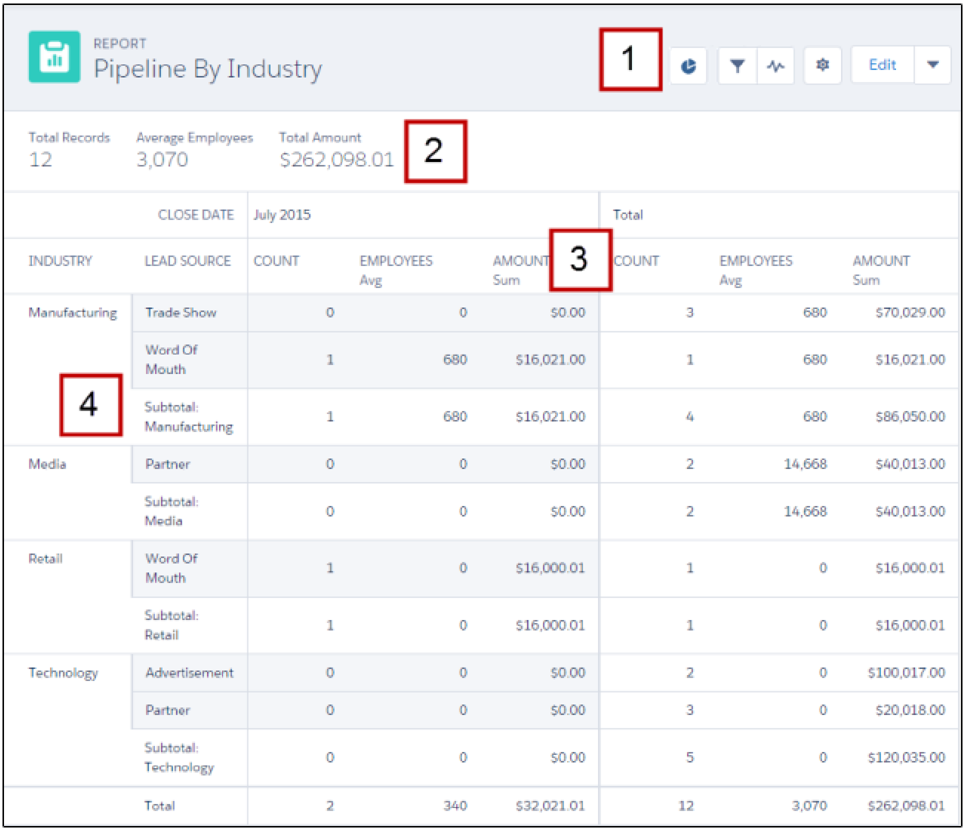
Record Count: 3

The buttons and drop-down menu (**1**) provide one-click access to administrative tasks, such as refresh, edit, clone, save, and delete.

Have more questions about a metric or chart? Click **View Report** (**2**) to drill into the data. Hover over a chart (**3**) to learn more about it.

Get More Information from Interactive Charts and Filters on the Report Run Page

In Lightning Experience, new features on the report run page ensure that reports answer even your toughest questions.



REPORT
Pipeline By Industry

Total Records 12 | Average Employees 3,070 | Total Amount $262,098.01

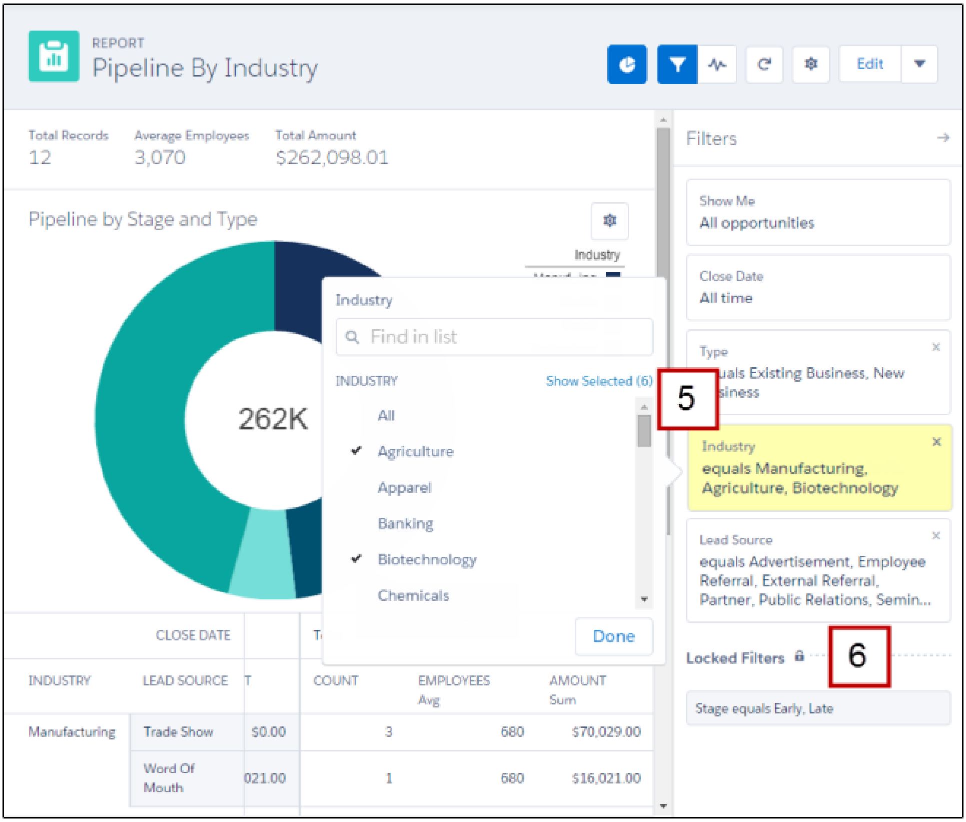
	CLOSE DATE	July 2015			Total		
INDUSTRY	LEAD SOURCE	COUNT	EMPLOYEES Avg	AMOUNT Sum	COUNT	EMPLOYEES Avg	AMOUNT Sum
Manufacturing	Trade Show	0	0	$0.00	3	680	$70,029.00
	Word Of Mouth	1	680	$16,021.00	1	680	$16,021.00
	Subtotal: Manufacturing	1	680	$16,021.00	4	680	$86,050.00
Media	Partner	0	0	$0.00	2	14,668	$40,013.00
	Subtotal: Media	0	0	$0.00	2	14,668	$40,013.00
Retail	Word Of Mouth	1	0	$16,000.01	1	0	$16,000.01
	Subtotal: Retail	1	0	$16,000.01	1	0	$16,000.01
Technology	Advertisement	0	0	$0.00	2	0	$100,017.00
	Partner	0	0	$0.00	3	0	$20,018.00
	Subtotal: Technology	0	0	$0.00	5	0	$120,035.00
	Total	2	340	$32,021.01	12	3,070	$262,098.01

Get more from your report with the tools in the header (**1**).

- ◔ — Show or hide a report chart.
- ▼ — Add, remove, or change report filters. After applying a filter, the report automatically refreshes to show filtered data. You no longer have to open the report builder to filter a report.
- ∿ — Open the report feed to collaborate with others on report data.
- ↻ — Refresh your report to show the latest data.
- ⚙ — Show or hide details like subtotals, grand totals, and record counts from your report.

Clicking **Edit** launches the report builder. The action menu provides one-click access to saving, cloning, and deleting the report.

You can view key metrics at the top of every report (**2**). Report headers float on both the X and Y axes (**3**), so you always know what field you're reviewing without needing to scroll. And, we've redesigned the report format so that groupings are easier to read (**4**).



Total Records 12 | Average Employees 3,070 | Total Amount $262,098.01

Industry
Find in list
INDUSTRY — Show Selected (6)
- All
- ✓ Agriculture
- Apparel
- Banking
- ✓ Biotechnology
- Chemicals

Done

Filters

Show Me: All opportunities

Close Date: All time

Type: uals Existing Business, New siness

Industry: equals Manufacturing, Agriculture, Biotechnology

Lead Source: equals Advertisement, Employee Referral, External Referral, Partner, Public Relations, Semin...

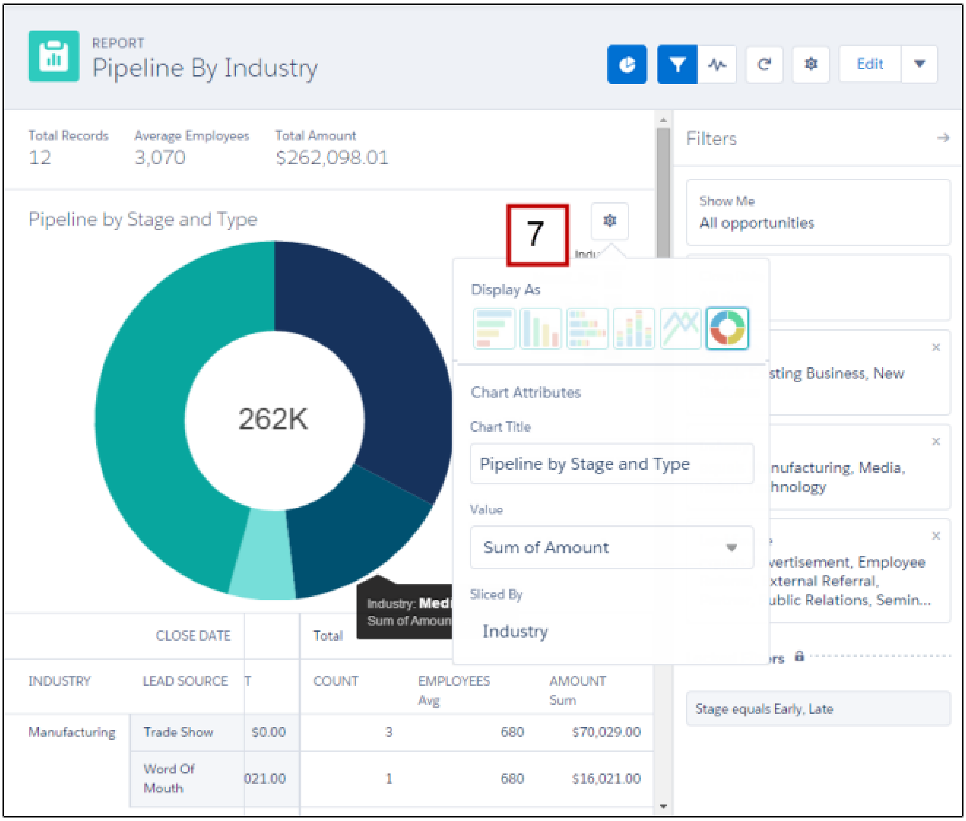
Locked Filters

Stage equals Early, Late

	CLOSE DATE		T		
INDUSTRY	LEAD SOURCE	T	COUNT	EMPLOYEES Avg	AMOUNT Sum
Manufacturing	Trade Show	$0.00	3	680	$70,029.00
	Word Of Mouth	021.00	1	680	$16,021.00

You can add, remove, and edit report filters (**5**) right from the report's page. You no longer have to open the report builder to filter the report you're reading. If you want to see your sales pipeline for the apparel industry, edit the `Industry` filter accordingly and your report refreshes.

You can also lock filters (**6**). If you want to share a report about your late-stage sales pipeline and don't want to share early-stage data, lock the `Stage` filter. Locked filters can't be edited on the report run page. You lock and unlock filters in the report builder.



Pipeline By Industry

Total Records 12 · Average Employees 3,070 · Total Amount $262,098.01

Pipeline by Stage and Type

Chart Title: Pipeline by Stage and Type

Value: Sum of Amount

Sliced By: Industry

Show Me: All opportunities

Stage equals Early, Late

		CLOSE DATE	Total		
INDUSTRY	LEAD SOURCE	T	COUNT	EMPLOYEES Avg	AMOUNT Sum
Manufacturing	Trade Show	$0.00	3	680	$70,029.00
	Word Of Mouth	021.00	1	680	$16,021.00

You're probably wondering: "But can I customize the chart?" Yes, you can (**7**)! Change the chart type, title, and more from the chart options menu (⚙).

Reports and Dashboards: Compatibility Between Lightning Experience and Salesforce Classic

Wondering how reports or dashboards created in one user interface work in the other? Here is a breakdown of what to expect.

For reports and dashboards created in Salesforce Classic:

- You can view and edit both in Lightning Experience.

- After saving a Salesforce Classic dashboard in Lightning Experience, you can't edit it in Salesforce Classic. Instead of editing a Salesforce Classic dashboard in Lightning Experience, consider cloning the dashboard and editing the clone. That way, you can still edit the original dashboard in Salesforce Classic.
- Lightning Experience obeys sharing rules set on report and dashboard folders in Salesforce Classic. Users can't set sharing rules by folder in Lightning Experience.
- You can view and open folders that you created in Salesforce Classic in Lightning Experience.
- Joined reports aren't supported in Lightning Experience.

For reports and dashboards created in Lightning Experience:

- You can view and edit reports created in Lightning Experience in Salesforce Classic.
- You can't edit dashboards created in Lightning Experience in Salesforce Classic, but you can view them.
- Dashboards that you create in Lightning Experience that have more than three columns automatically display in Salesforce Classic with three columns (retaining all dashboard components).

For more information on the limitations for reports and dashboards, see the Salesforce Help.

CHAPTER 9 Your Layouts Can Customize Lightning Experience Records

In this chapter ...

- Page Layouts in Lightning Experience
- Compact Layouts in Lightning Experience

You can customize the content of your record pages in Lightning Experience using tools you're already familiar with: page layouts and compact layouts.

Page Layouts in Lightning Experience

When you customize your object record pages in Salesforce Classic, those changes can affect the content of object record pages in Lightning Experience. However, in Lightning Experience, the page elements display differently, and some aren’t supported.

If you’re in an org that supports multiple page layouts, you can create a page layout directly from the Page Layouts related list on any object in the Object Manager. You can also edit or delete an object’s page layouts by clicking ▼ on a page layout in the Page Layouts related list.

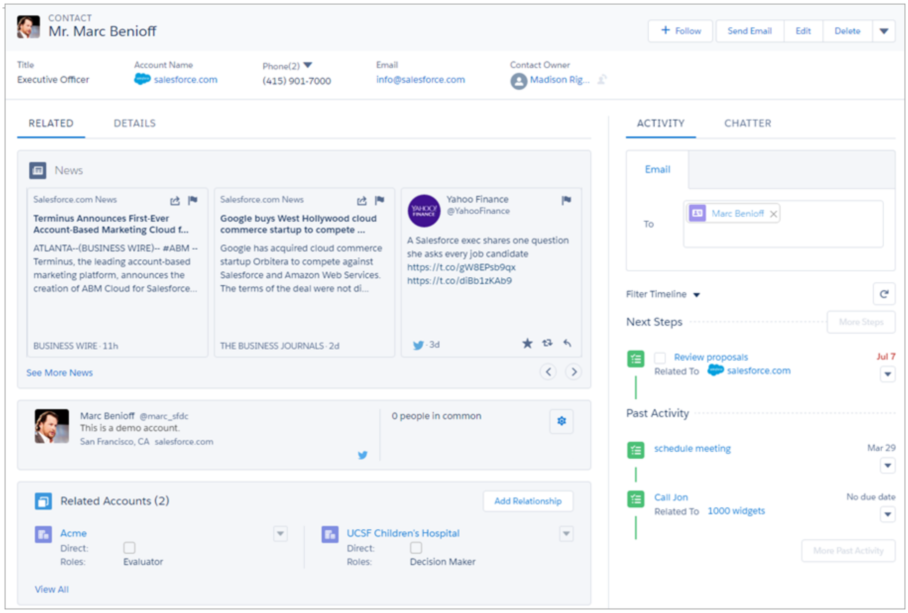
Here’s a sample contact record in Lightning Experience. The highlights panel, which contains key fields for the record, is the only part of a record page that you can’t customize using the page layout editor. The fields in the highlights panel are customized using a compact layout.

EDITIONS

Available in: Lightning Experience

Page layouts are available in: **All** Editions

Creation and deletion of page layouts is available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions



CONTACT
Mr. Marc Benioff

Title: Executive Officer
Account Name: salesforce.com
Phone(2): (415) 901-7000
Email: info@salesforce.com
Contact Owner: Madison Rig...

News

Salesforce.com News
Terminus Announces First-Ever Account-Based Marketing Cloud f...
ATLANTA--(BUSINESS WIRE)-- #ABM -- Terminus, the leading account-based marketing platform, announces the creation of ABM Cloud for Salesforce...
BUSINESS WIRE · 11h

Salesforce.com News
Google buys West Hollywood cloud commerce startup to compete ...
Google has acquired cloud commerce startup Orbitera to compete against Salesforce and Amazon Web Services. The terms of the deal were not di...
THE BUSINESS JOURNALS · 2d

Yahoo Finance @YahooFinance
A Salesforce exec shares one question she asks every job candidate https://t.co/gW8EPsb9qx https://t.co/dIBb1zKAb9
3d

Marc Benioff @marc_sfdc
This is a demo account.
San Francisco, CA salesforce.com
0 people in common

Related Accounts (2)

Acme
Direct: □
Roles: Evaluator

UCSF Children’s Hospital
Direct: □
Roles: Decision Maker

Email
To: Marc Benioff

Next Steps
□ Review proposals — Jul 7
Related To salesforce.com

Past Activity
schedule meeting — Mar 29
Call Jon — No due date
Related To 1000 widgets

These page layout elements are supported in Lightning Experience.

Actions

Actions display in different places, such as the highlights panel, Activity tab, and the Collaborate tab. The actions are derived from the list of actions in the Salesforce1 and Lightning Experience Actions section of the page layout. Some actions aren't supported in Lightning Experience.

For more information, see Actions in Lightning Experience on page 102.

Canvas Apps

Canvas apps are supported in Lightning Experience.

Custom Links

Custom links display under the Details tab.

Fields

Fields display under the Details tab. You can remove or reorder fields on a page layout only via the page layout editor.

Related Lists

Related lists are included as Lightning components in Lightning Experience. Not all related lists are supported in Lightning Experience. For example, the *`Object`* History related list isn't supported.

Standard and Custom Buttons

Standard and custom buttons are treated as actions in Lightning Experience, just like in Salesforce1.

Important: Custom buttons that call JavaScript aren't supported in Lightning Experience.

Visualforce Pages

Visualforce pages that you've added to the page layout appear under the Details tab. Only Visualforce pages with `Available for Salesforce mobile apps and Lightning Pages` enabled display in Lightning Experience.

Visualforce pages that have been put into the Mobile Cards section as components don't appear in Lightning Experience.

These page layout elements aren't supported in Lightning Experience.

- Blank spaces
- Expanded lookups
- Mobile cards

  Note: The Twitter card that you see on account, contact, and lead record pages in Lightning Experience isn't the same as the Twitter component available as a mobile card in the page layout editor. The Twitter card in Lightning Experience is a Lightning component. You must have Social Accounts and Contacts enabled for it to appear.

- S-controls

- Sections
- Tags

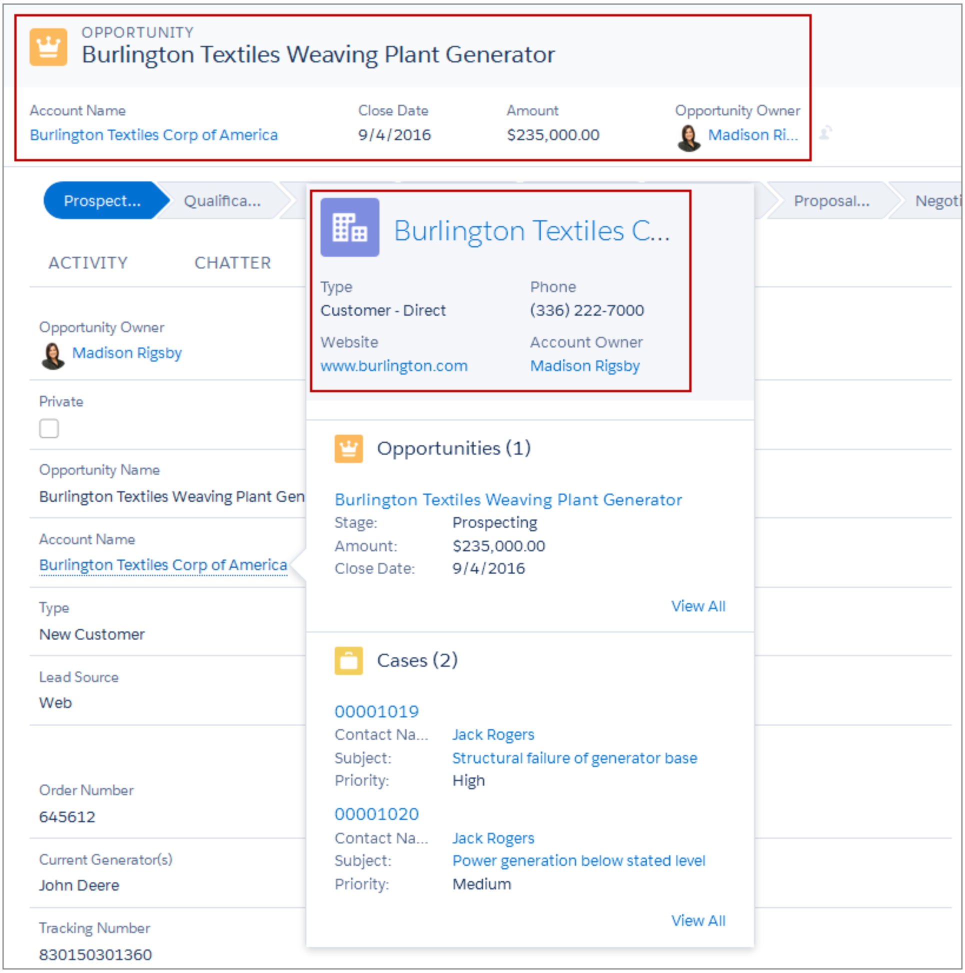
Note: You can't use the enhanced page layout editor to customize the layout of Lightning Experience record home pages.

Compact Layouts in Lightning Experience

If you've completed the Salesforce1 Mobile Basics module, you're familiar with compact layouts and how they work in Salesforce1. Compact layouts play the same role in Lightning Experience: displaying a record's key fields in the highlights panel of a record page.

A compact layout lets you put the most important fields at the top of a record page where your users can easily see them. If your org supports record types, you can assign a compact layout to different record types, just like you can with a page layout.

In Lightning Experience, the first five fields that you add to a compact layout display in an object's record highlights panel and in the expanded lookup card you see when you hover over a link in record details. The field you put first displays at the top in bold.



OPPORTUNITY
Burlington Textiles Weaving Plant Generator

Account Name	Close Date	Amount	Opportunity Owner
Burlington Textiles Corp of America	9/4/2016	$235,000.00	Madison Ri...

Opportunity Owner
Madison Rigsby

Private
□

Opportunity Name
Burlington Textiles Weaving Plant Gen

Account Name
Burlington Textiles Corp of America

Type
New Customer

Lead Source
Web

Order Number
645612

Current Generator(s)
John Deere

Tracking Number
830150301360

Burlington Textiles C...

Type	Phone
Customer - Direct	(336) 222-7000
Website	**Account Owner**
www.burlington.com	Madison Rigsby

Opportunities (1)

Burlington Textiles Weaving Plant Generator
Stage: Prospecting
Amount: $235,000.00
Close Date: 9/4/2016

View All

Cases (2)

00001019
Contact Na... Jack Rogers
Subject: Structural failure of generator base
Priority: High

00001020
Contact Na... Jack Rogers
Subject: Power generation below stated level
Priority: Medium

View All

Tip: Put the object's `Name` field first to provide context for your users when they view a record.

You can create and edit a compact layout from the Compact Layouts related list on any object in the Object Manager in Lightning Experience.

Changes you make to a compact layout are reflected in both Salesforce1 and Lightning Experience.

CHAPTER 10 Actions and Lightning Experience

In this chapter ...

- Actions in Lightning Experience
- How Actions Are Ordered in Lightning Experience

Actions enable users to do more in Salesforce, such as create or update records and log calls.

If you've already created and used actions in your organization, you're familiar with how they work in Salesforce Classic. If you've used our mobile apps, you've seen how they work in Salesforce1. In Lightning Experience, instead of showing up in one place—like the Chatter publisher or the Salesforce1 action bar—actions are split into different areas.

Next, we go over where you can find actions, which actions are and aren't supported, and how the customizations you've made to actions on a page layout affect how they display in Lightning Experience.

Actions in Lightning Experience

In Lightning Experience, actions display in the Global Actions menu in the header, on list view items, and in several places on a record page. Where they display on a record page depends on the action's type.

EDITIONS

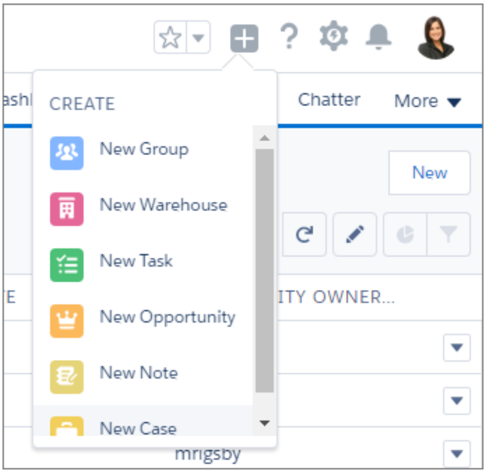
Available in: both Salesforce Classic and Lightning Experience

Quick actions available in: **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Contact Manager**, **Database.com**, and **Developer** Editions

Custom canvas actions available in: **Professional** (with Force.com Canvas enabled), **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Actions in the Global Actions Menu

The Global Actions menu displays a subset of global actions from the Salesforce1 and Lightning Experience Actions section of the global publisher layout.

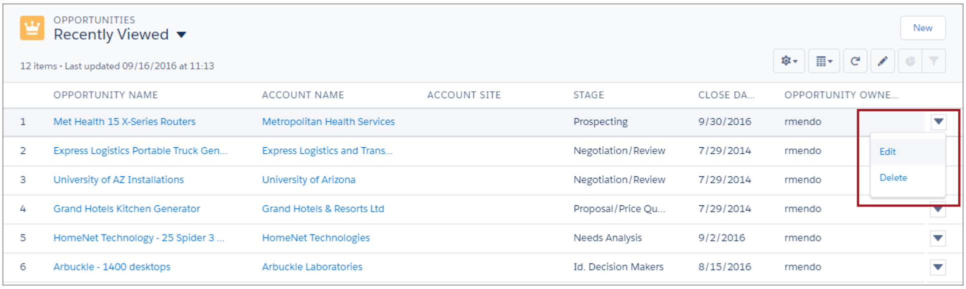


The items in the menu display in the order that they're listed in the Salesforce1 and Lightning Experience Actions section of the page layout.

Actions associated with objects that aren't supported in Lightning Experience don't display in the Global Actions menu. Also, the Global Actions menu doesn't support the standard Chatter actions Post, File, Poll, Link, Question, and Thanks.

Actions on List View Items

Except for the Tasks object, only standard button actions are supported on list view items. Items in Tasks list views contain the full list of actions available for tasks.



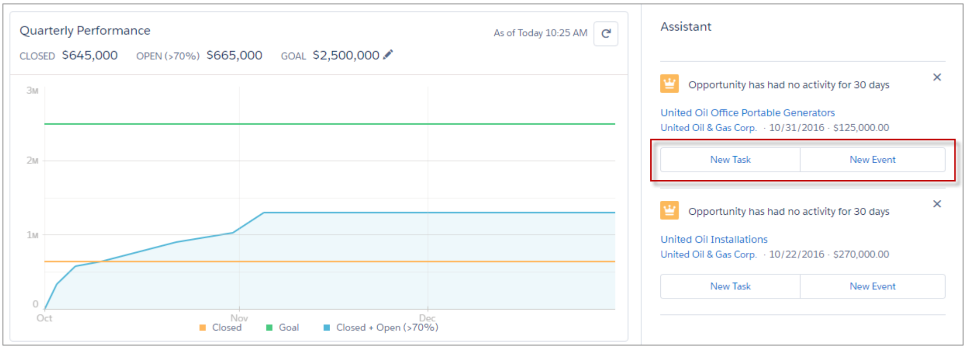
OPPORTUNITIES
Recently Viewed ▼

12 items • Last updated 09/16/2016 at 11:13

	OPPORTUNITY NAME	ACCOUNT NAME	ACCOUNT SITE	STAGE	CLOSE DA...	OPPORTUNITY OWNE...
1	Met Health 15 X-Series Routers	Metropolitan Health Services		Prospecting	9/30/2016	rmendo
2	Express Logistics Portable Truck Gen...	Express Logistics and Trans...		Negotiation/Review	7/29/2014	rmendo
3	University of AZ Installations	University of Arizona		Negotiation/Review	7/29/2014	rmendo
4	Grand Hotels Kitchen Generator	Grand Hotels & Resorts Ltd		Proposal/Price Qu...	7/29/2014	rmendo
5	HomeNet Technology - 25 Spider 3 ...	HomeNet Technologies		Needs Analysis	9/2/2016	rmendo
6	Arbuckle - 1400 desktops	Arbuckle Laboratories		Id. Decision Makers	8/15/2016	rmendo

Actions on the Home Page

On the Home page, you can find actions on recommendations in the Assistant. For example, if a sales rep receives an update that an opportunity doesn't have any open activity, the rep can create a task or event directly from the recommendation.



Quarterly Performance

As of Today 10:25 AM

CLOSED $645,000 OPEN (>70%) $665,000 GOAL $2,500,000

Assistant

Opportunity has had no activity for 30 days

United Oil Office Portable Generators
United Oil & Gas Corp. · 10/31/2016 · $125,000.00

Opportunity has had no activity for 30 days

United Oil Installations
United Oil & Gas Corp. · 10/22/2016 · $270,000.00

The actions that appear depend on the type of recommendation. The available actions include:

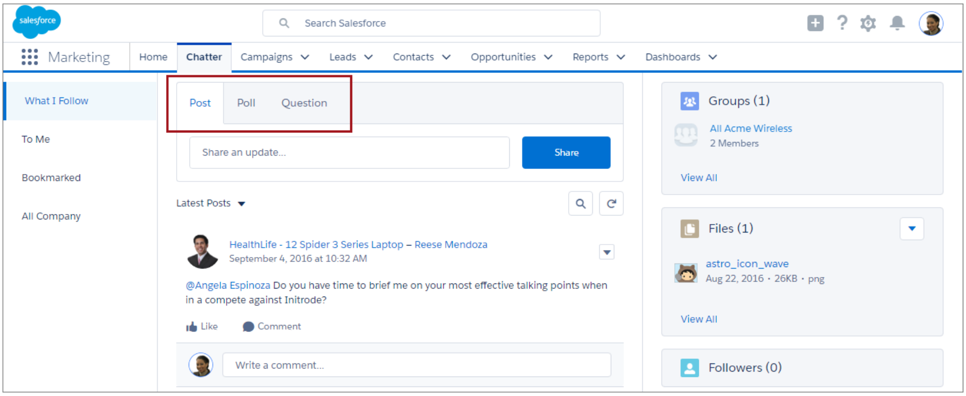
- New Task
- New Event
- Edit
- Email

After you complete an action, the related recommendation disappears from the Assistant.

Actions on the Chatter Page

The Chatter page, like the Chatter tab on record pages, contains only standard Chatter actions. By default, only the Post, Poll, and Question actions are supported, and if you have Groups, the Announcement action.

You can add, remove, or reorder the actions on the Chatter page from the Salesforce1 and Lightning Experience Actions section of the global publisher layout.

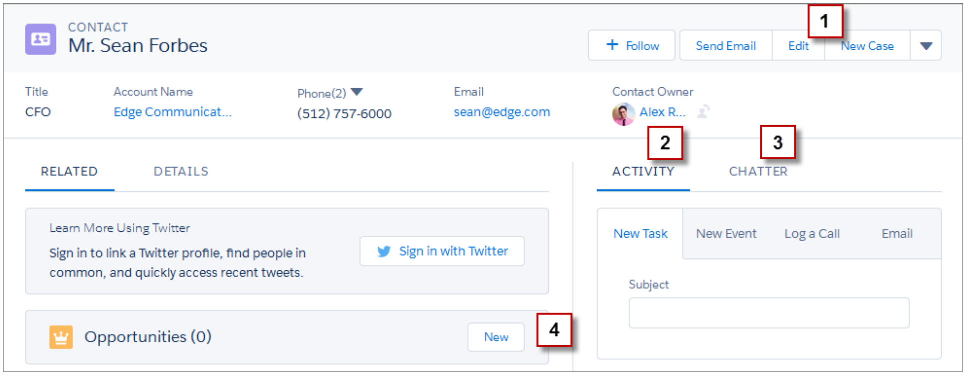


@Angela Espinoza Do you have time to brief me on your most effective talking points when in a compete against Initrode?

Actions on Record Pages

Here's a sample contact page in Lightning Experience.

Note: The opportunity and leads workspaces have different structures, but actions appear in the same way on those pages.



Sign in to link a Twitter profile, find people in common, and quickly access recent tweets.

The page-level action menu in the record's highlights panel (1) contains:

- Productivity actions
- Global and object-specific quick actions, except for those related to creating tasks, creating events, and logging calls

- Standard buttons
- Custom Lightning component quick actions
- Custom Visualforce quick actions
- Custom Visualforce buttons
- Canvas actions

The actions that appear in the page-level action menu display in the order that they are listed in the Salesforce1 and Lightning Experience Actions section of the page layout.

The Activity tab (2) contains Create a Record quick actions that point to the Event and Task objects. It also contains Log A Call actions and Send Email actions.

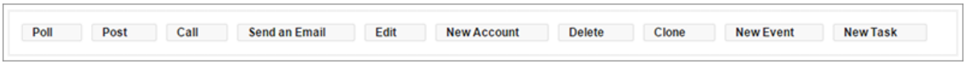
The Chatter tab (3) contains standard Chatter actions. By default, only the Post, Poll, and Question actions are supported, and if you have Groups, the Announcement action. Some objects support other standard Chatter actions predefined by Salesforce.

Note: Actions on cases and work orders appear in a different way than on other records. On case and work order records, the page-level action menu contains custom buttons and supported standard buttons. Quick actions appear on the Feed tab.

Actions on Related Lists

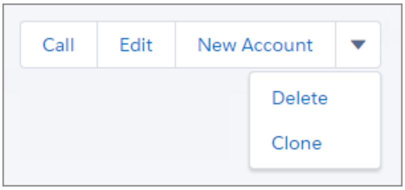
Related lists (4) contain the standard buttons normally found on related lists. Usually, it's simply the New button.

Example: Let's say you have these actions on your Contact page layout in the Salesforce1 and Lightning Experience Actions section.

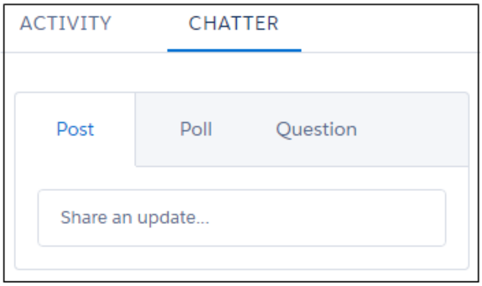


You have quick actions (New Account, New Event, New Task), a productivity action (Call), standard buttons (Edit, Delete, Clone, Send an Email), and Chatter actions (Poll, Post). Here's how those actions display on a contact record page in Lightning Experience.

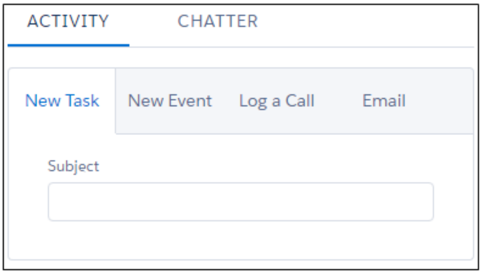
- The actions in the page-level action menu are a combination of the quick actions, productivity actions, and standard buttons in the order that they're listed on the page layout. Although they're quick actions, New Event and New Task don't show up here.



- The Chatter actions from the front of the action list are on the Chatter tab.



- The Activities-related actions—Email, New Event, New Task—display on the Activity tab.



How Actions Are Ordered in Lightning Experience

In Lightning Experience, the actions on record pages are derived from the list of actions in the Salesforce1 and Lightning Experience Actions section of the page layout for that object. The same section on global publisher layouts determines the global actions that appear in the Global Actions menu.

If you haven't customized the Salesforce1 and Lightning Experience Actions section of an object's page layout, the quick actions that appear on the object's record pages are derived from:

- The actions on the global publisher layout
- Standard and custom buttons in the buttons section of the object page layout

If you customize the Salesforce1 and Lightning Experience Actions section, the standard and custom buttons in the buttons section of the page layout aren't automatically included in the action list. You must add the buttons as actions from the Salesforce1 & Lightning Actions category in the palette.

The actions in each section of the record page respect the ordering of its types of actions on the page layout.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Quick actions available in: **Group**, **Professional**, **Enterprise**, **Performance**, **Unlimited**, **Contact Manager**, **Database.com**, and **Developer** Editions

Custom canvas actions available in: **Professional** (with Force.com Canvas enabled), **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

The Global Actions menu (+) in the Lightning Experience header displays all global quick actions from the Salesforce1 and Lightning Experience Actions section of the global publisher layout, except the standard Chatter actions Post, File, Poll, Link, Question, and Thanks.

CREATE LIGHTNING APPS

CHAPTER 11 What Is a Lightning App?

In this chapter ...

- Navigate at the Speed of Lightning
- Find Items and Apps in the App Launcher
- Meet the Lightning Experience App Manager
- Install a Custom Classic App: The Warehouse Data Model

An *app* is a collection of items that work together to serve a particular function. Salesforce apps come in two flavors: Classic apps and Lightning apps. Classic apps are created and managed in Salesforce Classic. Lightning apps are created and managed in Lightning Experience. You can customize both types of app to match the way your users work.

Similar to apps in Salesforce Classic, Lightning apps give your users access to sets of objects, tabs, and other items all in one convenient bundle in the navigation bar. However, Lightning apps take things to a level beyond Classic apps. Lightning apps let you brand your apps with a custom color and logo. You can even include a utility bar and Lightning page tabs in your Lightning app.

With apps in Lightning Experience, members of your org can work more efficiently by easily switching between apps. Users can open apps you've created from the App Launcher. What's most important to sales reps? Accounts, events, and organizations. How about sales managers? Reports and dashboards make the top of the list.

Let's jump into the details.

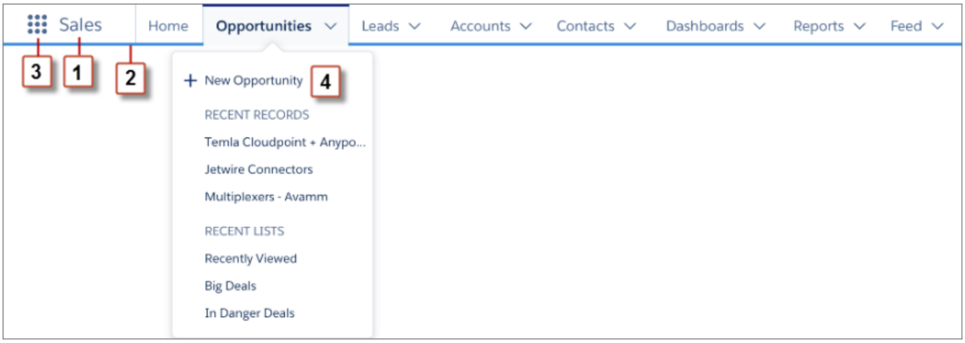
Navigate at the Speed of Lightning

If you know Salesforce Classic, the Lightning Experience navigation model will feel like a familiar friend, only better.

Each Lightning app has a navigation bar at the top of the page, letting your users:

- Find what they need using item names for easy recognition
- Complete actions and access recent records and lists with a single click

Think of the navigation bar as a container for a set of items and functionality. It's always there, but the items within it change based on the app you're using.



- The app name displays on the left side of the navigation bar (1) and custom colors and branding (2) make each app unique and easy to identify.
- Your users can access other items and apps by clicking the App Launcher icon (3) or the app name.
- Your users can create records and access recent records and lists directly from the navigation bar (4) for items like Opportunities.

The items in an app are always just one click away in the navigation bar.

What can you put in the Lightning app navigation bar?

If you're familiar with Salesforce Classic, you know that Classic apps can contain:

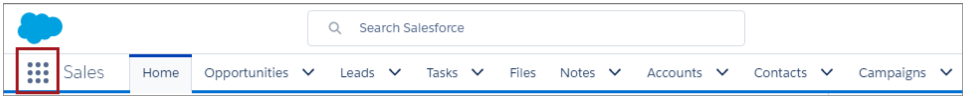
- Most standard objects, including Home, the main Chatter feed, Groups, and People
- Your org's custom objects
- Visualforce tabs
- Lightning component tabs

- Canvas apps via Visualforce tabs
- Web tabs

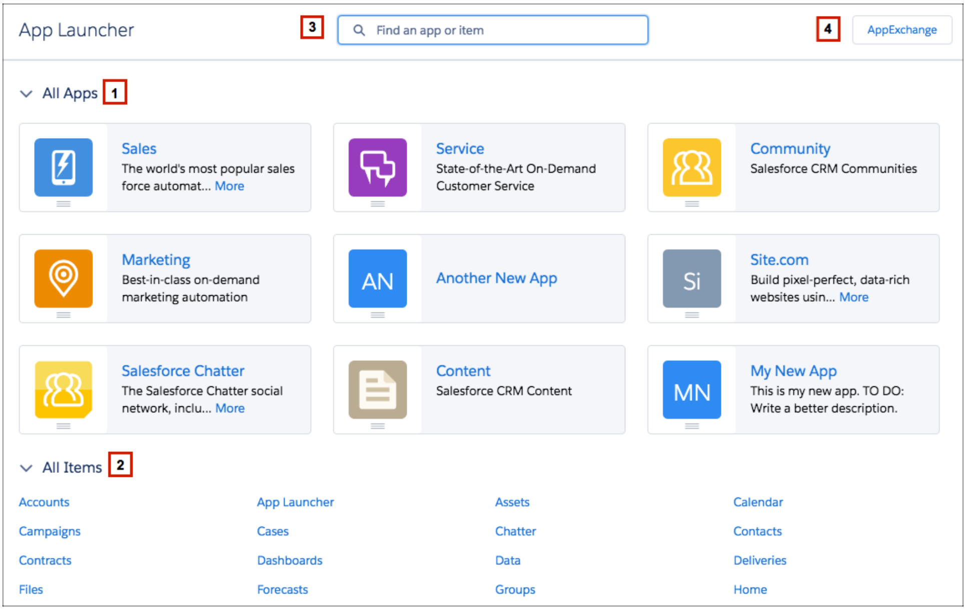
Lightning apps can contain everything on this list plus Lightning Page tabs and utilities like Lightning Voice. If your org uses utility features, you can enable a utility bar in your app that allows instant access to productivity tools, like integrated voice, in the Lightning Experience footer.

Find Items and Apps in the App Launcher

With the App Launcher, your users can switch between apps and access available items and features. The App Launcher is so central to navigating Lightning Experience and its apps, we've positioned it within easy reach on the left side of the navigation bar. It's available from any page.



The App Launcher displays all available apps in one place so your users can easily find what they're looking for—even apps they didn't know they had.

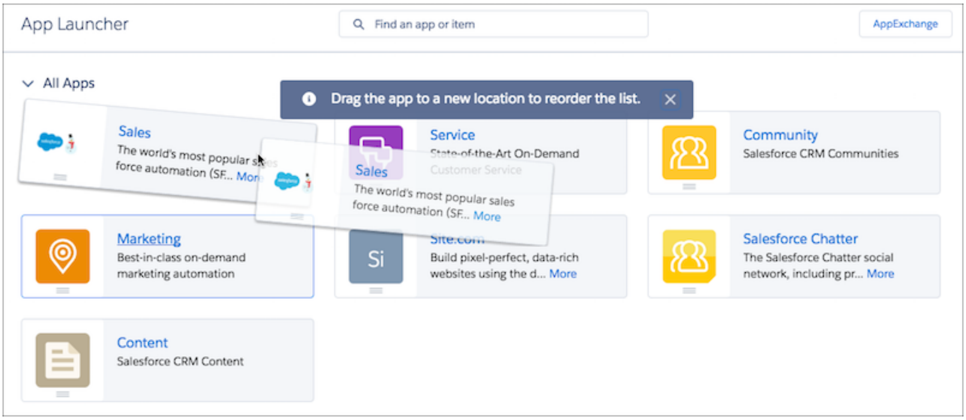

App Launcher
3
Find an app or item
4
AppExchange
All Apps
1
Sales
The world's most popular sales force automat... More
Service
State-of-the-Art On-Demand Customer Service
Community
Salesforce CRM Communities
Marketing
Best-in-class on-demand marketing automation
AN
Another New App
Si
Site.com
Build pixel-perfect, data-rich websites usin... More
Salesforce Chatter
The Salesforce Chatter social network, inclu... More
Content
Salesforce CRM Content
MN
My New App
This is my new app. TO DO: Write a better description.
All Items
2
Accounts
App Launcher
Assets
Calendar
Campaigns
Cases
Chatter
Contacts
Contracts
Dashboards
Data
Deliveries
Files
Forecasts
Groups
Home

Apps show up in the App Launcher as large tiles under All Apps (1). Other items, such as custom objects, tasks, events, and the feed, show up under All Items (2). Don't see what you want? Search for it by name in the search box (3). Expand your search to find the latest cloud-computing apps and services available on the AppExchange (4). You can do it all with the click of a button without ever leaving Lightning Experience.

You can make various types of apps available to your users in the App Launcher, including:

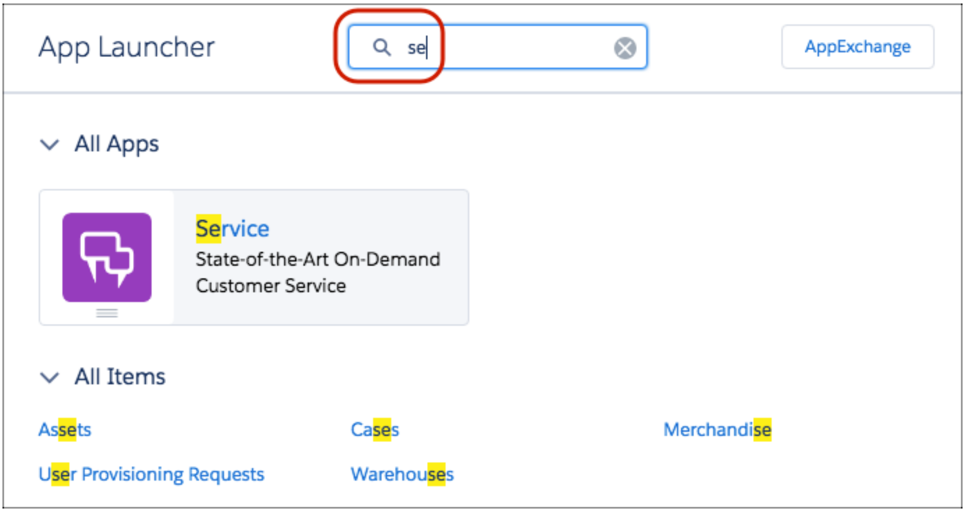
- Salesforce apps, which include custom apps and standard apps that come with Salesforce, like Sales and Service
- Connected apps such as Gmail™ and Microsoft® Office 365™
- Partner and ISV apps

As an admin, you can change which apps appear on the App Launcher and the default order in which they appear. You and your users can then drag the tiles around to create your own personal view of the App Launcher. Your users see only the apps you authorize for them to access through profiles or permission sets.


App Launcher
Find an app or item
AppExchange
All Apps
Sales
The world's most popular sales force automation (SF... More
Service
State-of-the-Art On-Demand Customer Service
Community
Salesforce CRM Communities
Marketing
Best-in-class on-demand marketing automation
Si
Site.com
Build pixel-perfect, data-rich websites using the d... More
Salesforce Chatter
The Salesforce Chatter social network, including pr... More
Content
Salesforce CRM Content

Drag the app to a new location to reorder the list.

The App Launcher's great for finding an app or item even when it's not currently on the navigation bar. Just click the App Launcher icon (⠿) to search for it by name. For example, say you're looking for an item called Service. Enter `Service` in the search box to see items and apps that match your search as you type.



All Apps

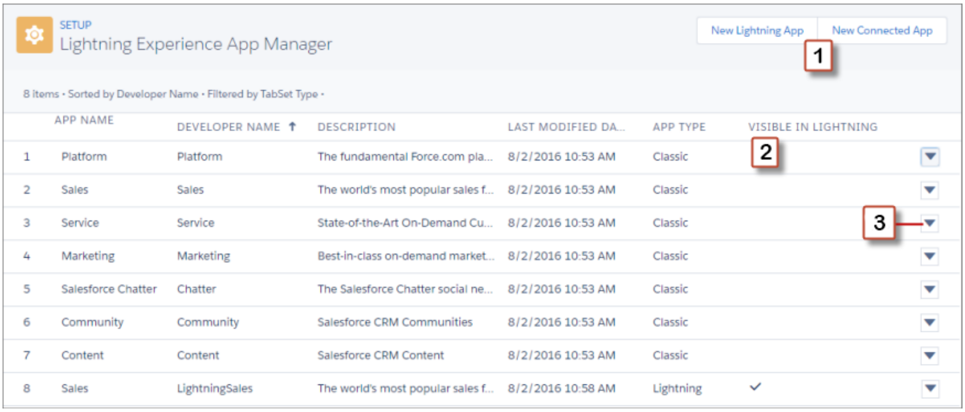
Service
State-of-the-Art On-Demand Customer Service

All Items

Assets	Cases	Merchandise
User Provisioning Requests	Warehouses	

Meet the Lightning Experience App Manager

The App Manager is your go-to place for managing apps for Lightning Experience. It shows all your connected apps and Salesforce apps, both Classic and Lightning.



8 items • Sorted by Developer Name • Filtered by TabSet Type •

	APP NAME	DEVELOPER NAME ↑	DESCRIPTION	LAST MODIFIED DA...	APP TYPE	VISIBLE IN LIGHTNING
1	Platform	Platform	The fundamental Force.com pla...	8/2/2016 10:53 AM	Classic	
2	Sales	Sales	The world's most popular sales f...	8/2/2016 10:53 AM	Classic	
3	Service	Service	State-of-the-Art On-Demand Cu...	8/2/2016 10:53 AM	Classic	
4	Marketing	Marketing	Best-in-class on-demand market...	8/2/2016 10:53 AM	Classic	
5	Salesforce Chatter	Chatter	The Salesforce Chatter social ne...	8/2/2016 10:53 AM	Classic	
6	Community	Community	Salesforce CRM Communities	8/2/2016 10:53 AM	Classic	
7	Content	Content	Salesforce CRM Content	8/2/2016 10:53 AM	Classic	
8	Sales	LightningSales	The world's most popular sales f...	8/2/2016 10:58 AM	Lightning	✓

You can use the Lightning Experience App Manager to:

- Create Lightning apps or connected apps (1)

- See if your Classic apps are accessible to your users in Lightning Experience (2)
- Edit, delete, or upgrade Classic apps to take advantage of all the benefits of apps in Lightning Experience (3)

Tip: Click a column header to sort the list based on that column.

What does that "Visible in Lightning" column mean?

A checkmark in the Visible in Lightning column means that the app is accessible in Lightning Experience via the App Launcher and is fully functional.

You can toggle the availability of your Classic apps in Lightning Experience by selecting or deselecting `Show in Lightning Experience` on the Classic app's detail page. Classic apps that have been upgraded to Lightning apps automatically have that setting disabled.

We've mentioned a couple of times now that you can upgrade your Classic apps. Why do that? So your apps can take advantage of all the perks of Lightning apps. We'll get there shortly. But first, let's get your org ready for it.

Install a Custom Classic App: The Warehouse Data Model

We want to walk you through upgrading a Classic app to Lightning. To do that, you need to have a clean custom Classic app in your org. So we're going to enlist an old friend for help: the enhanced Warehouse data model. You might be familiar with the Warehouse app if you've gone through the tutorials in the *Force.com Workbook*. We took the Warehouse data from that guide and added a few extra things to it. If you've gone through the exercises in the *Salesforce1 Developer Guide*, you probably already have the enhanced Warehouse app installed.

Important: Check your Developer org. If you already have the Warehouse app installed, skip these steps.

1. Go to `www.salesforce.com` and log in to Salesforce using your Developer Edition credentials.
2. Open a new browser tab or window, and navigate to `https://github.com/forcedotcom/Salesforce1-Dev-Guide-Setup-Package`.

   Do this in the browser that you logged in to your Developer org with.
3. Open the README file.
4. Click the link in the README file.

This is the installation link for the enhanced Warehouse data package. You should be taken directly into your development org to the Package Installation Details page.

5. Click **Install**.
6. Wait for the installation to finish.
7. From the Force.com app menu in Salesforce Classic, select Warehouse.
8. Click the Data tab.
9. Click **Create Data**.
10. Click your name in the upper-right corner, then select **Switch to Lightning Experience**.

CHAPTER 12 Step into the World of Lightning Apps

In this chapter ...

- Create a Lightning App
- Tips for Creating Apps in Lightning Experience

Creating and editing a Lightning app is a cinch. As in Salesforce Classic, you can create apps in Lightning Experience, but with even more bells and whistles. You can brand and customize Lightning apps to help your users work more efficiently.

For example, you can create a Lightning app for your finance department that includes all important items (including tabs) that users need to complete common tasks. You can customize the navigation bar color, brand it with a logo, and make the app available in the App Launcher for the user profiles associated with the finance department.

Tip: If your org includes utilities like Lightning Voice or Open CTI Softphone, you can add them to your app's utility bar.

Create a Lightning App

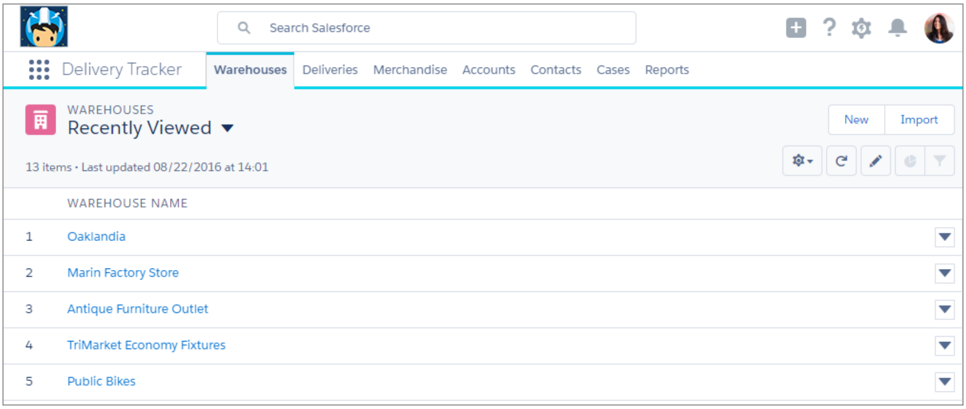
You've got the Warehouse app installed in your org. Let's create a mini app for the warehouse delivery managers, who want to know what inventory is getting delivered where, whether it got there on time, and what cases related to deliveries are open.

1. From Setup, enter *`App`* in the `Quick Find` box, then select **App Manager**.
2. Click **New Lightning App**.
3. In the Lightning App Wizard, create an app with these parameters.

App name	Delivery Tracker
Description	Track warehouse deliveries.
Image	Your choice! (We chose Astro.) Use a JPG, PNG, BMP, or GIF image that's smaller than 5 MB. For best results, upload an image that's 128 by 128 pixels. Images larger than the maximum display of 128 by 128 pixels are automatically resized.
Primary hex color value	#09D4EA
App navigation	Standard
Navigation bar items (in this order)	Warehouses, Deliveries, Merchandise, Accounts, Contacts, Cases, Reports
Assigned to user profile	System Administrator

4. Click **Save and Finish** to exit the wizard.
5. Click ⋮⋮⋮ to open the App Launcher, and select the Delivery Tracker app.
6. Check out the new app!

   It's got all the custom branding we gave it: a custom icon in the upper left and the custom color we assigned to it. Because Warehouses is first in the navigation bar, it becomes the landing page for the app. And even though App Launcher was an item available for the navigation bar, we didn't have to add it. It's there, accessible by clicking ⋮⋮⋮ .



WAREHOUSES
Recently Viewed ▼

13 items • Last updated 08/22/2016 at 14:01

	WAREHOUSE NAME
1	Oaklandia
2	Marin Factory Store
3	Antique Furniture Outlet
4	TriMarket Economy Fixtures
5	Public Bikes

Nice work! Now you're ready to create your own custom Lightning apps.

Beyond the Basics

Did you know that app images for Lightning apps can be animated GIFs? Oh yes, they can. You're welcome.

Tips for Creating Apps in Lightning Experience

EDITIONS

Available in: Lightning Experience

Available in: **Enterprise**, **Professional**, **Performance**, **Unlimited**, and **Developer** Editions

It's time for the fun part: deciding how to set up Lightning apps for your users. Here are some tips for planning Lightning apps for your org.

The best time to create Lightning apps is when you're rolling out Lightning Experience. So make creating Lightning apps a part of your rollout strategy. Check out the Trailhead module "Lightning Experience Rollout" for many great ideas to help you make a smooth transition.

Talk to your users. Ask them what their priorities are. Customizing tabs in apps gives you a unique opportunity to engage with your users. Each group of users has its own priorities. Find out which objects and items represent their highest priorities.

- Ask users to post feedback to a Chatter group.
- Publish polls.

- Schedule lunch sessions. Everyone likes a free lunch, and nearly everybody is happy to express their opinion.

Create a master list of objects that everyone in your org wants. Then trim down the list for each group—sales reps, sales managers, execs, and so on. The menus for every user group share some common objects, like Home, Tasks, and Feed. Keep the high-priority items for each group at the top. Put low-priority items at the bottom, or remove them altogether. Users can always go to the App Launcher to get the items they use less often.

CHAPTER 13 Take Your Classic App to the Next Level: Lightning

In this chapter ...

- Upgrade the Warehouse App
- Give the Warehouse App That Touch of Lightning
- Look at the Warehouse App Now

Classic apps work in Lightning Experience. So why upgrade them to Lightning apps? Upgrading a Classic app to a Lightning app lets you and your users take advantage of custom branding and the enhanced navigation features available in Lightning Experience.

You installed the enhanced Warehouse app into your org, but it installed as a Classic app. Let’s bring it into Lightning.

Upgrade the Warehouse App

Even though a Classic app works in Lightning Experience, it doesn't take advantage of all the benefits of being a Lightning app. That's why we recommend that you upgrade it. Let's do that now.

1. From Setup, enter *`App`* in the `Quick Find` box, then select **App Manager**.
2. Find the Warehouse app in the list.
3. Click ▼ from the Warehouse app row, and select **Upgrade**.
4. Leave the suggested "Warehouse Lightning" name as-is, and click **Upgrade**.
   Your Classic app is copied and upgraded for Lightning Experience. You now have two versions of the app: a Classic version, and a Lightning version. After you upgrade it, the Classic app is no longer accessible in Lightning Experience via the App Launcher. You'll still see the Classic app in the apps list, but with the Visible in Lightning column deselected.

After you upgrade a Classic app, the two versions of the app must be managed separately. Future changes you make to the Classic app won't be reflected in the Lightning version of the app, and vice versa.

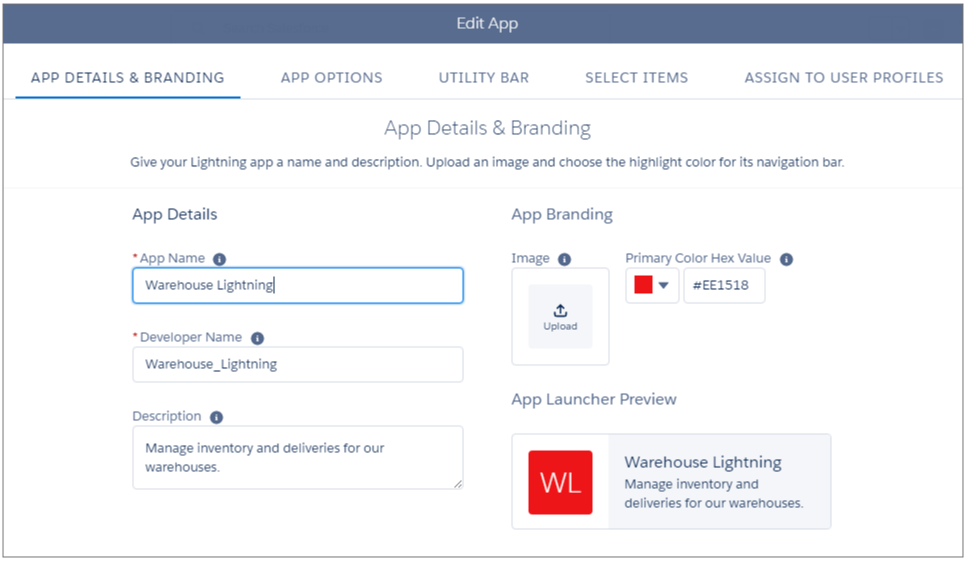
Give the Warehouse App That Touch of Lightning

You've upgraded the Warehouse app, but it's not taking advantage of the great enhancements that Lightning apps can offer. Let's fix that.

1. Find the Warehouse Lightning app in the list.
2. Click ▼ from the Warehouse Lightning app row, and select **Edit**.
3. Update the description to *`Manage inventory and deliveries for our warehouses.`*

   The app description displays alongside the icon in the App Launcher, so it's a great way to tell your users what the app is for, but keep it brief.
4. In the App Branding section, change the Primary Color Hex Value to *`#EE1518`*.

   You can see a preview of the default app icon using the red color we assigned and what the app tile will look like in the App Launcher.



App Details & Branding

Give your Lightning app a name and description. Upload an image and choose the highlight color for its navigation bar.

App Details

*App Name: Warehouse Lightning

*Developer Name: Warehouse_Lightning

Description: Manage inventory and deliveries for our warehouses.

App Branding

Image

Primary Color Hex Value: 🟥 #EE1518

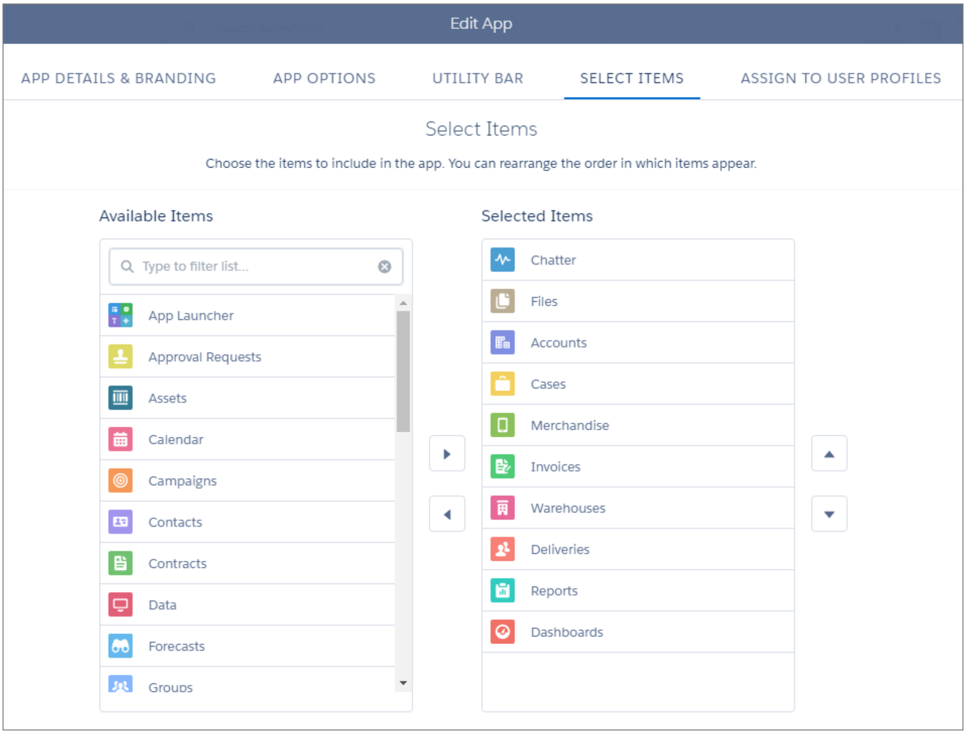
App Launcher Preview

WL Warehouse Lightning
Manage inventory and deliveries for our warehouses.

5. Click **Save**.

6. Click the Select Items tab, and remove the Data and Home tabs from the Selected Items list. That leaves the Chatter tab at the top of the list, which means it becomes the app's landing page.

7. Add Cases to the Selected Items list, and move it to directly after Accounts.



Select Items

Choose the items to include in the app. You can rearrange the order in which items appear.

Available Items

Type to filter list...

- App Launcher
- Approval Requests
- Assets
- Calendar
- Campaigns
- Contacts
- Contracts
- Data
- Forecasts
- Groups

Selected Items

- Chatter
- Files
- Accounts
- Cases
- Merchandise
- Invoices
- Warehouses
- Deliveries
- Reports
- Dashboards

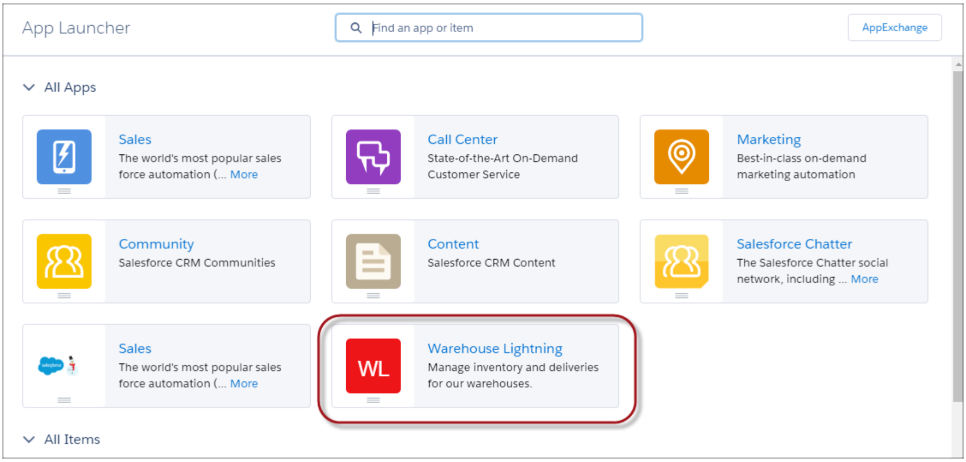
Tip: If you have Lightning pages in your org, you can see them in the Available Items list. Being able to add Lightning pages to the navigation bar is another great perk about Lightning apps.

8. Click **Save**.
9. Click the Assign to User Profiles tab, and make sure that System Administrator is in the Selected Items list.
10. Click **Done**.

Look at the Warehouse App Now

Let's look at the changes we made to the Warehouse app.

1. Click ⋮⋮⋮ from the app navigation bar.
   There's the Warehouse Lightning app.


App Launcher
Find an app or item
AppExchange
All Apps
Sales
The world's most popular sales force automation (... More
Call Center
State-of-the-Art On-Demand Customer Service
Marketing
Best-in-class on-demand marketing automation
Community
Salesforce CRM Communities
Content
Salesforce CRM Content
Salesforce Chatter
The Salesforce Chatter social network, including ... More
Sales
The world's most popular sales force automation (... More
WL
Warehouse Lightning
Manage inventory and deliveries for our warehouses.
All Items

2. Click the Warehouse Lightning app.
   If the “Welcome to Salesforce” Setup Assistant dialog appears, close it.

   You’re in the updated Warehouse app. It’s got Chatter as the landing page, Cases right after Accounts, and the bright red color we gave it.

Good job! You’ve got the skills. Now go create and update some apps in your own org.

CREATE CUSTOM PAGES FOR LIGHTNING EXPERIENCE

CHAPTER 14 Introduction to the Lightning App Builder

In this chapter ...

- How the Lightning App Builder Works
- The Lightning App Builder User Interface
- Lightning Page Types

Your users are busy. They're closing deals, providing top-notch service, and marketing to your prospects and customers. By creating customized pages, you can put key information at your users' fingertips and provide them with an easy interface to update and add records.

The Lightning App Builder is a point-and-click tool that makes it easy to create custom pages for Salesforce1 and Lightning Experience. With the Lightning App Builder, you can combine various components on a single page to give your users what they need all in one place.

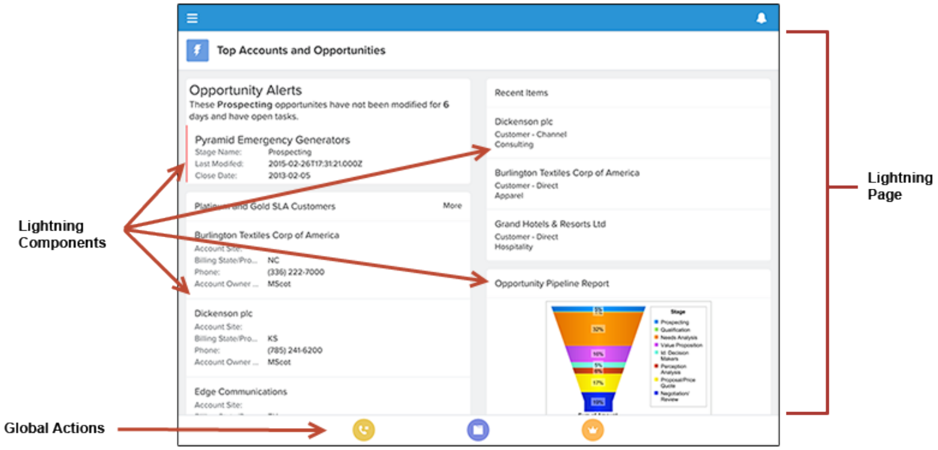
How the Lightning App Builder Works

With the Lightning App Builder, you can build:

- Single-page apps that drill down into standard pages
- Dashboard-style apps, such as apps to track top sales prospects or key leads for the quarter
- "Point" apps to solve a particular task, such as an expense app for users to enter expenses and monitor expenses they've submitted
- Custom record pages for your objects, tailored to the needs of your users
- Custom Home pages containing the components and features that your users use most

Lightning Pages are the underlying technology for the Lightning App Builder. A Lightning Page is a custom layout that lets you design pages for use in the Salesforce1 mobile app or in Lightning Experience. A Lightning Page is composed of regions that contain components.

Here's a sneak peek at one of the pages you're going to build.



Top Accounts and Opportunities

Opportunity Alerts

These **Prospecting** opportunites have not been modified for **6** days and have open tasks.

Pyramid Emergency Generators
Stage Name: Prospecting
Last Modifed: 2015-02-26T17:31:21.000Z
Close Date: 2013-02-05

Platinum and Gold SLA Customers More

Burlington Textiles Corp of America
Account Site:
Billing State/Pro... NC
Phone: (336) 222-7000
Account Owner ... MScot

Dickenson plc
Account Site:
Billing State/Pro... KS
Phone: (785) 241-6200
Account Owner ... MScot

Edge Communications
Account Site:

Recent Items

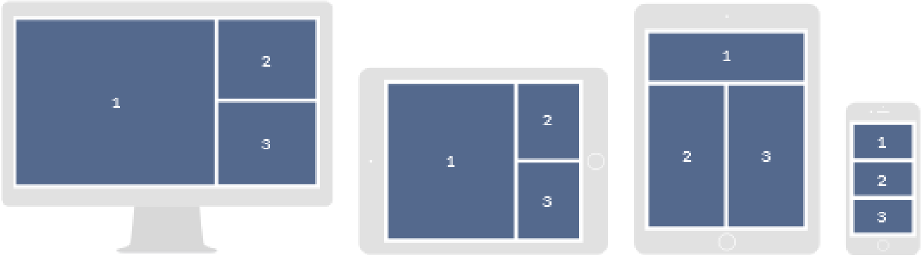
Dickenson plc
Customer - Channel
Consulting

Burlington Textiles Corp of America
Customer - Direct
Apparel

Grand Hotels & Resorts Ltd
Customer - Direct
Hospitality

Opportunity Pipeline Report

The structure of a Lightning Page adapts for the device it's viewed on. The template you choose when creating the page controls how it displays on a given device. The Lightning Page's template divides the page into regions.



Lightning Components

A Lightning component is a compact, configurable, and reusable element that you can drag and drop onto a Lightning Page in the Lightning App Builder.

You can use standard, custom, and third-party components in the Lightning App Builder.

Standard Components

Standard components are Lightning components built by Salesforce.

Custom Components

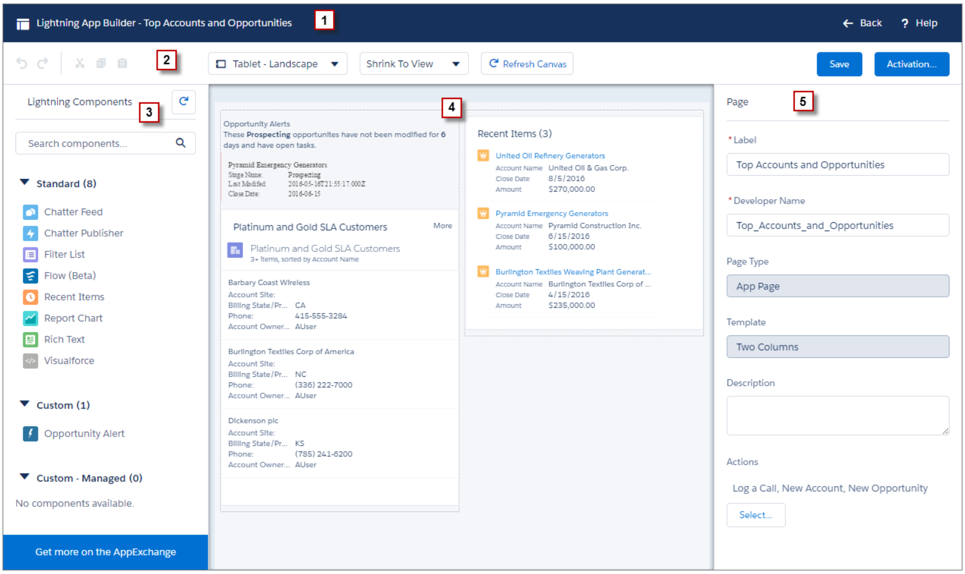
Custom components are Lightning components that you or someone else have created. With some modifications, custom Lightning components can work in the Lightning App Builder. For more information on making Lightning components App Builder–friendly, see the *Lightning Components Developer Guide*.

Third-Party Components on AppExchange

The AppExchange provides a marketplace for Lightning components. You can find packages containing components already configured and ready to use in the Lightning App Builder.

The Lightning App Builder User Interface

The Lightning App Builder's user interface makes creating Lightning Pages easy. Here's a breakdown of the parts of the tool.



Opportunity Alerts
These **Prospecting** opportunities have not been modified for **6** days and have open tasks.

Pyramid Emergency Generators
Stage Name: Prospecting
Last Modified: 2016-05-16T21:55:17.000Z
Close Date: 2016-06-15

Platinum and Gold SLA Customers
3+ items, sorted by Account Name

Barbary Coast Wireless
Account Site:
Billing State/Pr... CA
Phone: 415-555-3284
Account Owner... AUser

Burlington Textiles Corp of America
Account Site:
Billing State/Pr... NC
Phone: (336) 222-7000
Account Owner... AUser

Dickenson plc
Account Site:
Billing State/Pr... KS
Phone: (785) 241-6200
Account Owner... AUser

Recent Items (3)

United Oil Refinery Generators
Account Name United Oil & Gas Corp.
Close Date 8/5/2016
Amount $270,000.00

Pyramid Emergency Generators
Account Name Pyramid Construction Inc.
Close Date 6/15/2016
Amount $100,000.00

Burlington Textiles Weaving Plant Generat...
Account Name Burlington Textiles Corp of ...
Close Date 4/15/2016
Amount $235,000.00

*Label: Top Accounts and Opportunities
*Developer Name: Top_Accounts_and_Opportunities
Page Type: App Page
Template: Two Columns
Description:
Actions: Log a Call, New Account, New Opportunity

Header (1)

The header shows you the label of your Lightning Page. You can also return to Setup without saving or to view more help for the Lightning App Builder.

Toolbar (2)

Use the buttons in the toolbar to cut (✂), copy (📋), and paste (📋) page content; and to undo (↺), redo (↻), save, or activate your Lightning Page. You can also view your page in different formats, refresh the canvas, or adjust the canvas size to fit your view.

Lightning Components Pane (3)

The components pane contains all standard and custom Lightning components that are supported for your Lightning Page. Click and drag a component to add it to the page.

Tip: If you have a lot of custom components, enter text in the search field to easily find the one you need. You can access third-party custom components on the AppExchange using the button at the bottom of the pane.

Lightning Page Canvas (4)

The canvas area is where you build your page. Drag components to reorder them on the page.

Properties Pane (5)

Depending on what you select on the page, the properties pane shows either the overall page properties or the properties of the component that you've selected. Click **Page** in the breadcrumb to access the page properties when viewing a component.

Lightning Page Types

You can create different types of Lightning Pages with the Lightning App Builder.

App Page

Use an app page to create a home page for a third-party app that you can add directly into the Salesforce1 and Lightning Experience navigation menus. Your users then have an app home page where they can quickly access the most important objects and items.

Home Page

Create Home pages with features relevant to specific types of users, and assign the customized pages to different user profiles. Custom Home pages are supported in Lightning Experience only.

Record Page

With a record page, you can create a customized version of an object's record page, tailoring it to your users' needs. Custom record pages are supported in Lightning Experience only.

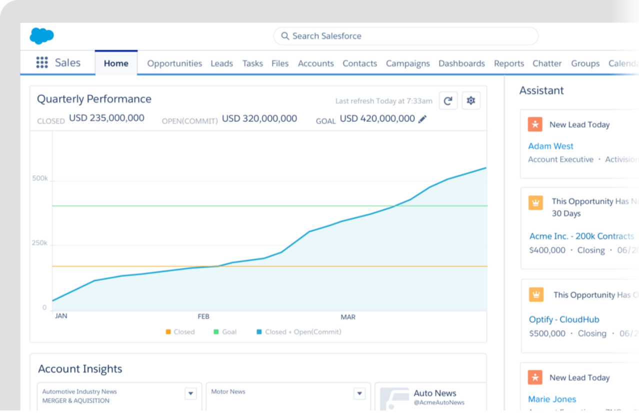
We'll create all three. Let's start with the Home page.

CHAPTER 15 Yes, Virginia, You Can Customize Your Lightning Experience Home Page

In this chapter ...

- Create a Custom Home Page
- Roll Out Your Custom Home Page to Your Lightning Experience Users
- Look at Your Handiwork

Give your users everything they need to manage their day from the Home page in Lightning Experience. Your sales reps can see their quarterly performance summary and get important updates on critical tasks and opportunities. You can also customize the page for different types of users and assign custom pages for different profiles.



You can create a custom Home page in two ways: Create it from scratch using a template, or edit an existing page.

To edit an existing page, you can click ⚙ from the Home page, and then select **Edit Page** to create a copy of the current page to edit. If a customized page exists and is active, selecting Edit Page opens that page to edit.

We're going to create a Home page from scratch.

Beyond the Basics

When you select Edit Page for the first time, Salesforce makes a copy of the standard page. This copy is what you then edit in the Lightning App Builder. Pages created like this—as copies—retain a reference to the standard page, which means that Salesforce can upgrade the page copies for you with new capabilities in the future. If you create a page from scratch, then you own it completely and new page capabilities Salesforce rolls out won't appear automatically.

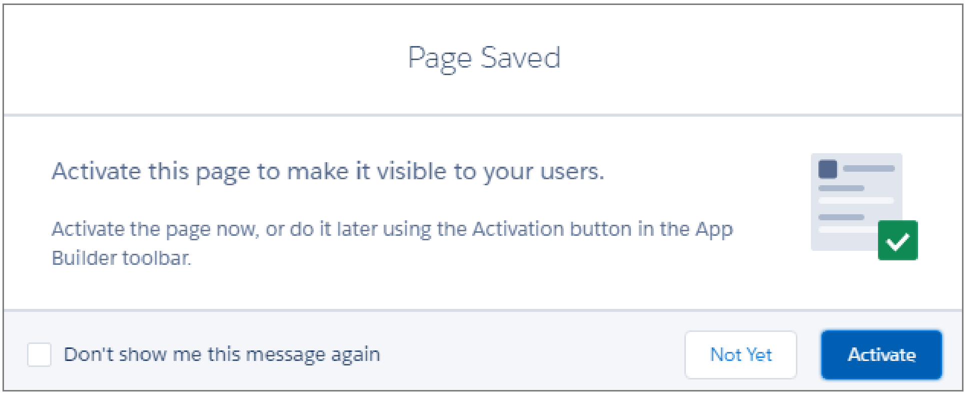
Create a Custom Home Page

We'll tweak the position of the components on the standard Home page layout slightly to give you an idea of what's possible.

Let's get started!

1. From Setup, enter `App Builder` in the `Quick Find` box, then select **Lightning App Builder**.
2. Click **New**, and select **Home Page**.
3. Step through the wizard and name the page `New Home Page`, select the standard home page template, and then click **Finish**.

   The components pane contains all the standard components available for the Home page.
4. Drag the Assistant component to the top right region.
5. Drag Quarterly Performance to the top left region.
6. Add Upcoming Events to the lower left region and add Top Deals to the lower right region.
7. Add Today's Tasks above the Assistant.
8. Click **Save**.



Page Saved

Activate this page to make it visible to your users.

Activate the page now, or do it later using the Activation button in the App Builder toolbar.

But wait, what's this? There's more? Yes, there is. Saving the page isn't enough to get it out to your users. Lightning pages must be activated before your users can see them.

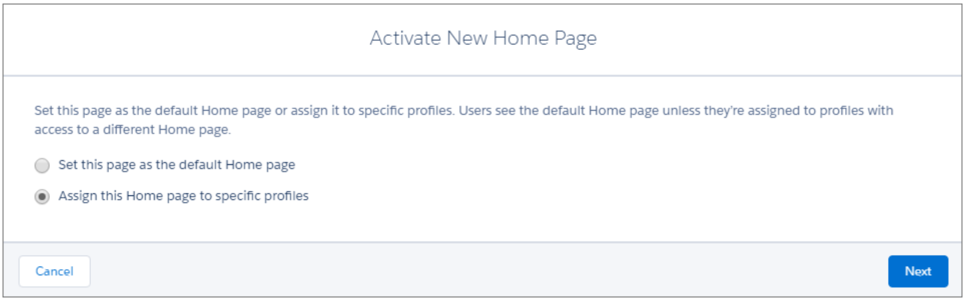
Normally, if you aren't done with your page, or aren't ready to make it public, you can click **Not Yet** here to save the page and return to the App Builder. But that's not us. We're bold! We're done with our page and want to give it to our users right now!

9. Click **Activate**, and we'll do just that.

   Tip: If you saved previously and didn't activate the page, you can click the **Activation** button in the toolbar to be ready for the next section.

Roll Out Your Custom Home Page to Your Lightning Experience Users

When activating a Home page, you have two options: You can make it the default for everyone, or assign it to one or more profiles, giving your users access to a page designed just for their role.



Activate New Home Page

Set this page as the default Home page or assign it to specific profiles. Users see the default Home page unless they're assigned to profiles with access to a different Home page.

○ Set this page as the default Home page

● Assign this Home page to specific profiles

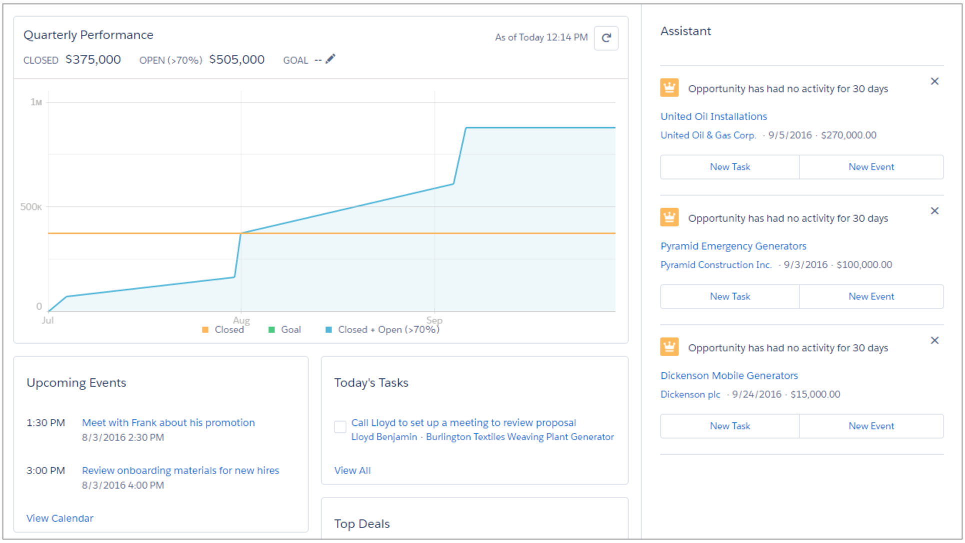
Let's assign this home page to the System Administrator profile so we can go look at it afterward.

1. Select `Assign this Home page to specific profiles`, then click **Next**.
2. Scroll down the list of profiles and select `System Administrator`.
3. Click **Next**, review the assignment, and then click **Activate**.

That's it. Let's go take a look.

Look at Your Handiwork

Just for reference, here's an example of the standard default Home page. Think of this as the "before" in this Home page makeover.



Quarterly Performance

As of Today 12:14 PM

CLOSED $375,000 OPEN (>70%) $505,000 GOAL --

Upcoming Events

1:30 PM Meet with Frank about his promotion
8/3/2016 2:30 PM

3:00 PM Review onboarding materials for new hires
8/3/2016 4:00 PM

View Calendar

Today's Tasks

□ Call Lloyd to set up a meeting to review proposal
Lloyd Benjamin · Burlington Textiles Weaving Plant Generator

View All

Top Deals

Assistant

Opportunity has had no activity for 30 days
United Oil Installations
United Oil & Gas Corp. · 9/5/2016 · $270,000.00
New Task | New Event

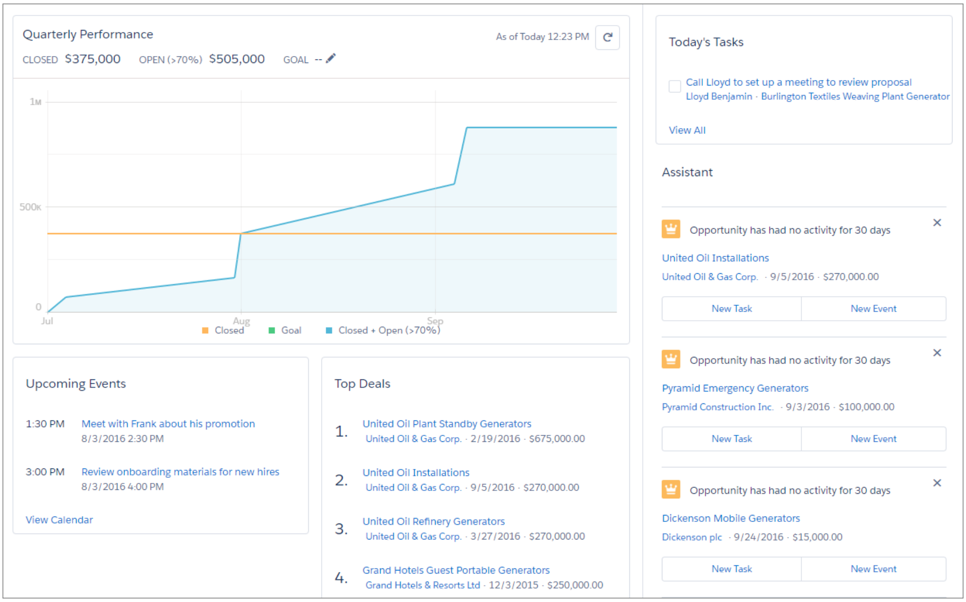
Opportunity has had no activity for 30 days
Pyramid Emergency Generators
Pyramid Construction Inc. · 9/3/2016 · $100,000.00
New Task | New Event

Opportunity has had no activity for 30 days
Dickenson Mobile Generators
Dickenson plc · 9/24/2016 · $15,000.00
New Task | New Event

Now, let's compare it to what we did.

1. Navigate to the Lightning Experience home page.

   Here's what the Home page looks like now:



Quarterly Performance

As of Today 12:23 PM

CLOSED $375,000 OPEN (>70%) $505,000 GOAL --

Upcoming Events

1:30 PM Meet with Frank about his promotion
8/3/2016 2:30 PM

3:00 PM Review onboarding materials for new hires
8/3/2016 4:00 PM

View Calendar

Top Deals

1. United Oil Plant Standby Generators
   United Oil & Gas Corp. · 2/19/2016 · $675,000.00
2. United Oil Installations
   United Oil & Gas Corp. · 9/5/2016 · $270,000.00
3. United Oil Refinery Generators
   United Oil & Gas Corp. · 3/27/2016 · $270,000.00
4. Grand Hotels Guest Portable Generators
   Grand Hotels & Resorts Ltd · 12/3/2015 · $250,000.00

Today's Tasks

□ Call Lloyd to set up a meeting to review proposal
Lloyd Benjamin · Burlington Textiles Weaving Plant Generator

View All

Assistant

Opportunity has had no activity for 30 days
United Oil Installations
United Oil & Gas Corp. · 9/5/2016 · $270,000.00
New Task | New Event

Opportunity has had no activity for 30 days
Pyramid Emergency Generators
Pyramid Construction Inc. · 9/3/2016 · $100,000.00
New Task | New Event

Opportunity has had no activity for 30 days
Dickenson Mobile Generators
Dickenson plc · 9/24/2016 · $15,000.00
New Task | New Event

2. Bask in the glory of your newfound page customization skills.

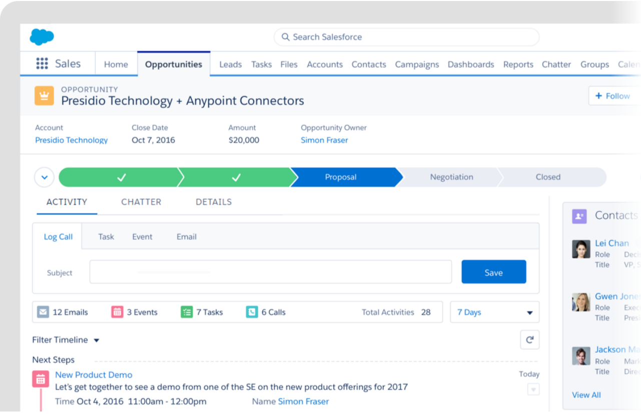
All right! Let’s move on to record pages.

CHAPTER 16 Those Lightning Experience Record Pages? You Can Customize Them, Too!

In this chapter ...

- Create a Custom Lightning Record Page
- Roll Out Your Custom Record Page to Your Lightning Experience Users
- See What You Did There?

Use the Lightning App Builder to add, remove, or reorder components on a record page to give users a customized view for each object's records. Even cooler, you can customize a record page and assign it to specific Lightning apps to give your users access to a unique record page customized especially for the context of the app they're working in.



Account: Presidio Technology
Close Date: Oct 7, 2016
Amount: $20,000
Opportunity Owner: Simon Fraser

New Product Demo

Let's get together to see a demo from one of the SE on the new product offerings for 2017

Time Oct 4, 2016 11:00am - 12:00pm Name Simon Fraser

Just like the Home page, you can create a custom record page in two ways: Create it from scratch using a template, or edit an existing page. We're going to create a record page using a template.

Create a Custom Lightning Record Page

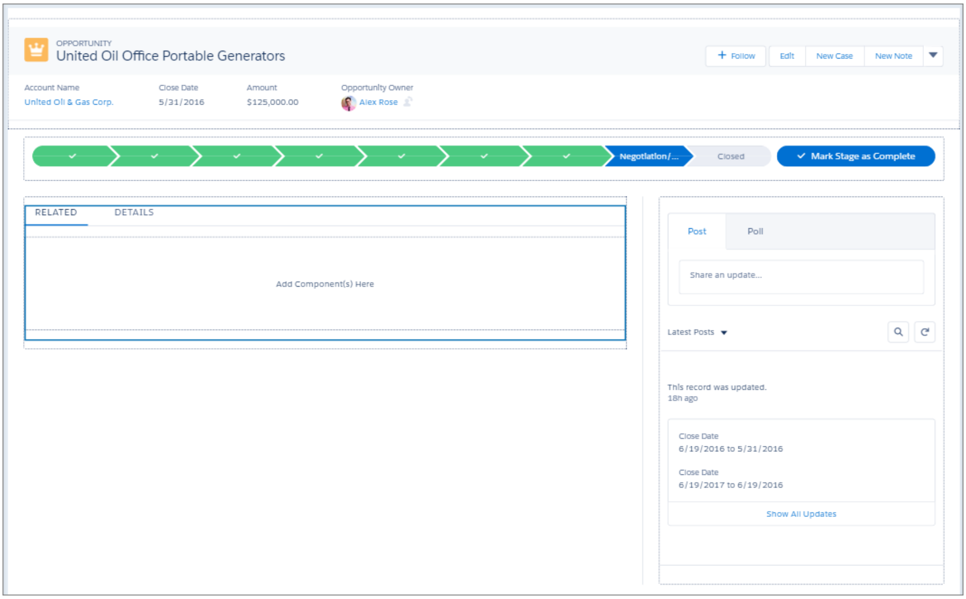
Let's build a custom opportunity record page from scratch.

We'll tweak the standard record page layout just a bit, so you can get a feel for how things go together. Once you're comfortable with that, you can go to town and customize your record pages any way you like. Let's get started.

1. From Setup, enter *`App Builder`* in the `Quick Find` box, then select **Lightning App Builder**.
2. Click **New**.
3. Select **Record Page**.
4. Name your page *`New Opportunity Page`*, and select **Opportunity**.
5. Choose the **Header, Subheader, Right Sidebar** template on the next screen, and click **Finish**.

   In the components pane, you see all the standard components available for opportunity record pages and any custom components that you've installed in your org.
6. Drag the Highlights Panel component into the top region of the page.

   Click **See How It Works** in the component properties pane to find out where the highlights panel content comes from.
7. Add the Path component to the region below the highlights panel.
8. Add a Chatter component to the lower right region.
9. Add a Tabs component to the lower left region.



Account Name: United Oil & Gas Corp.
Close Date: 5/31/2016
Amount: $125,000.00
Opportunity Owner: Alex Rose

This record was updated.
18h ago

Close Date
6/19/2016 to 5/31/2016

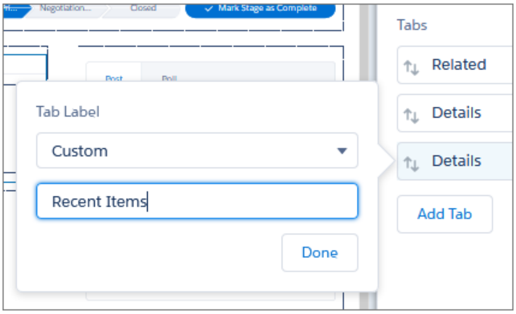
Close Date
6/19/2017 to 6/19/2016

The Tabs component comes with some default tabs already in place. Let's add one more.

10. In the Tabs component details pane, click **Add Tab**.

    By default, another Details tab is added. But since we already have one, let's change this new one to something else.

11. Click the second Details tab. From the Tab Label drop-down menu, select **Custom**, and give the tab a new label: `Recent Items`.



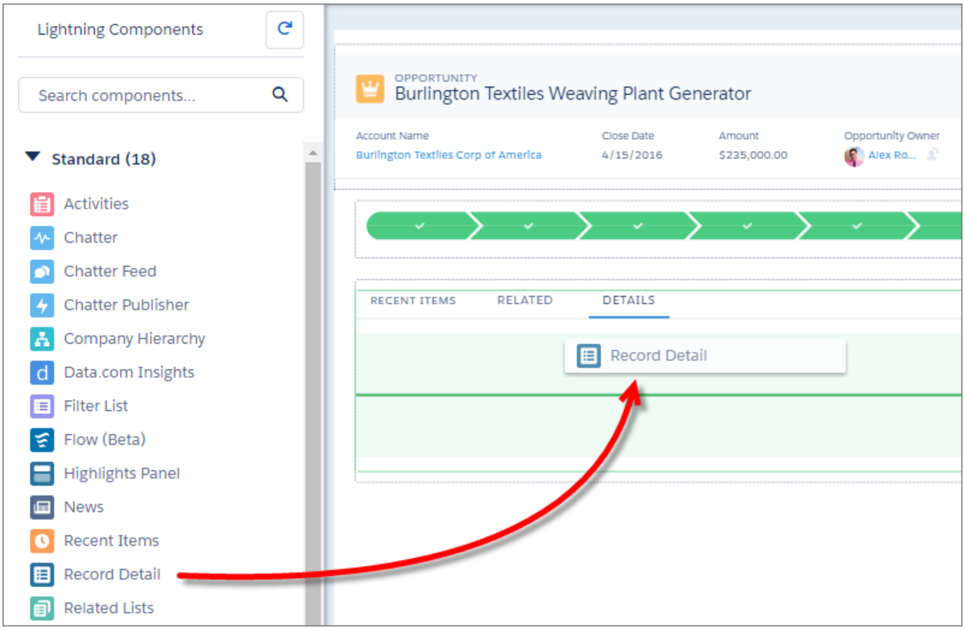
Tab Label: Custom
Recent Items

12. Click **Done**, and drag the Recent Items tab to the top of the Tabs list in the properties pane.

The Recent Items tab is now in the first position in the tabs component. You can click around between the tabs, but nothing changes because the tabs don't have any components assigned to them. They're empty. Let's fix that.

To add the first component to a tab, select the tab on the canvas and then drop the component directly below it.

13. Select the Details tab.
14. Drag a Record Detail component right below the Details tab, into the green highlighted area.



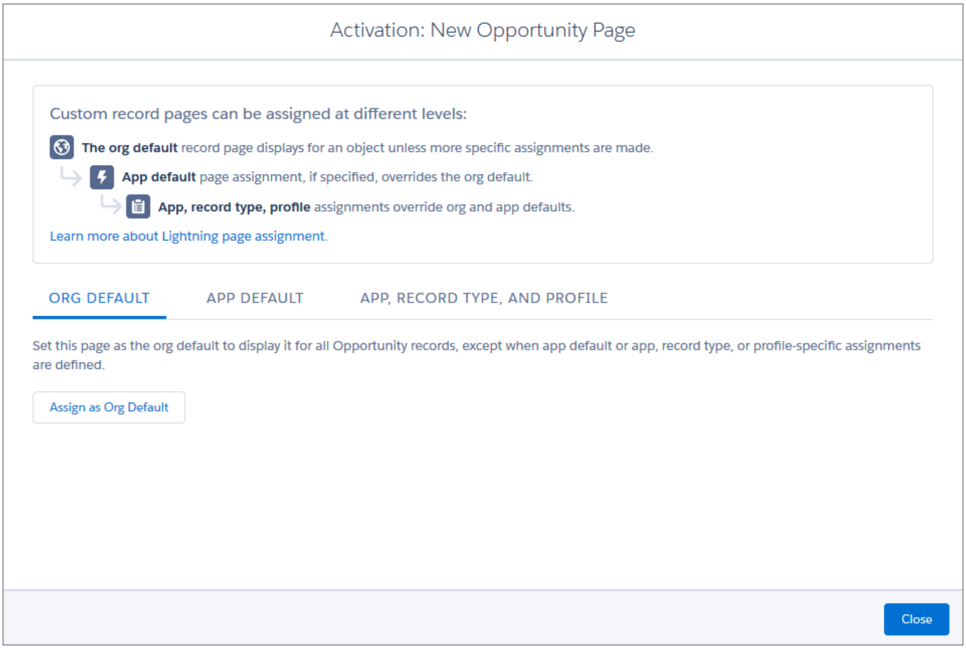
15. Create an Activity tab.
16. Add a Related Lists component to the Related tab, an Activities component to the Activity tab, and the Recent Items component to the Recent Items tab.
17. Select the tabs component on the canvas, and in the properties pane, change the order of the tabs to: Details, Activity, Recent Items, and Related.

    You can't drag the tabs inside the component to move them around. You can only adjust them in the properties pane.

18. Click **Save**, then **Activate**.

Roll Out Your Custom Record Page to Your Lightning Experience Users

It's time to spread the awesomeness! Let's activate the page. It's super easy.



Activation: New Opportunity Page

Custom record pages can be assigned at different levels:

The org default record page displays for an object unless more specific assignments are made.

App default page assignment, if specified, overrides the org default.

App, record type, profile assignments override org and app defaults.

Learn more about Lightning page assignment.

Set this page as the org default to display it for all Opportunity records, except when app default or app, record type, or profile-specific assignments are defined.

You have three options for activation.

- Make the page the org default for the object.
- Make the page the default object record page for specific Lightning apps.
- Assign the page to a combination of Lightning apps, record types, and profiles.

Let's assign this page to a specific app, record type, and profile.

1. Click the App, Record Type, and Profile tab.
2. Click **Assign to Apps, Record Types, and Profiles**.
3. Assign the page to the Sales app, the Master record type, and the System Administrator profile.
4. Review the page assignment.

The New Page column is populated with the name of the page we're activating: New Opportunity Page.

5. Click **Save**.

See? Easy peasy. Your customized record page is now live. Let's go check it out.

Tip: You may be thinking, "This is great, but what if I change my mind? How do I de-activate my custom page?" That's easy, too. Click **Activation**, click the App, Record Types, and Profile tab, and click **Remove Assignments**.

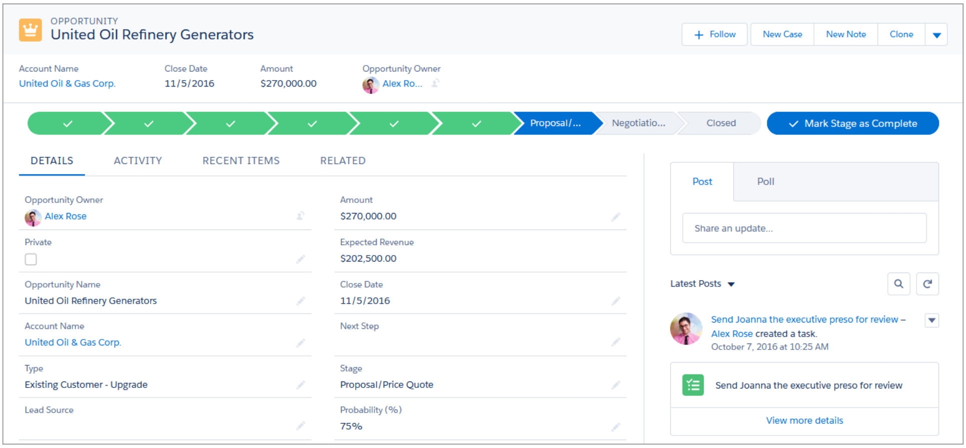
See What You Did There?

You've created a page and activated it. Now let's see it in action.

1. Click **Back** in the App Builder header.
2. From the App Launcher (⠿), open the Sales app, and click the Opportunities tab.
3. Select any opportunity.

   You might have to refresh the opportunity page for the record page changes to show up.

   Here's what the custom record page looks like, using the sample company, United Oil Refinery Generators. Because we assigned the record page to the System Administrator profile, you can see it, but no other users in your org can. You can customize your different users' experiences by creating custom record pages and assigning them by app, record type, and user profile. Give your sales managers a different view of opportunities than your sales reps. Configure nonprofit account pages differently than standard business account pages.



OPPORTUNITY
United Oil Refinery Generators

Account Name: United Oil & Gas Corp.
Close Date: 11/5/2016
Amount: $270,000.00
Opportunity Owner: Alex Ro...

Opportunity Owner: Alex Rose
Amount: $270,000.00
Private: □
Expected Revenue: $202,500.00
Opportunity Name: United Oil Refinery Generators
Close Date: 11/5/2016
Account Name: United Oil & Gas Corp.
Next Step:
Type: Existing Customer - Upgrade
Stage: Proposal/Price Quote
Lead Source:
Probability (%): 75%

Share an update...

Latest Posts

Send Joanna the executive preso for review – Alex Rose created a task.
October 7, 2016 at 10:25 AM

Send Joanna the executive preso for review

You did it! You've taken your first steps into a larger world.

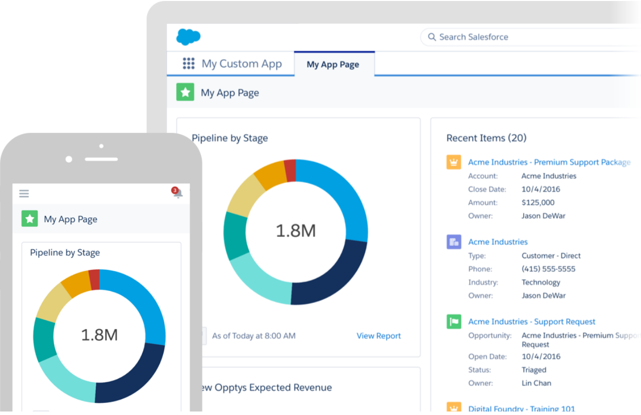
Let's move on to app pages.

CHAPTER 17 The App Home Lightning Page

In this chapter ...

- Create an App Page
- Add Your App Page to Salesforce1 and Lightning Experience
- Test Your App Page in Salesforce1
- Test Your App Page in Lightning Experience

Add a custom home page for an app to Salesforce1 and Lightning Experience to let your users easily access the objects and items that are most important in that app.



Acme Industries - Premium Support Package
Account: Acme Industries
Close Date: 10/4/2016
Amount: $125,000
Owner: Jason DeWar

Acme Industries
Type: Customer - Direct
Phone: (415) 555-5555
Industry: Technology
Owner: Jason DeWar

Acme Industries - Support Request
Opportunity: Acme Industries - Premium Suppo Request
Open Date: 10/4/2016
Status: Triaged
Owner: Lin Chan

Digital Foundry - Training 101

Create an App Page

Let's build an app home page for a sales team.

Your sales team needs to see top deals in the pipeline, with a visual interface that makes it easy to absorb key details at a glance. They want to see the most recent opportunities they've viewed and be able to drill into the record details with a single tap or click. And they want functionality to log calls and create accounts and opportunities on the go.

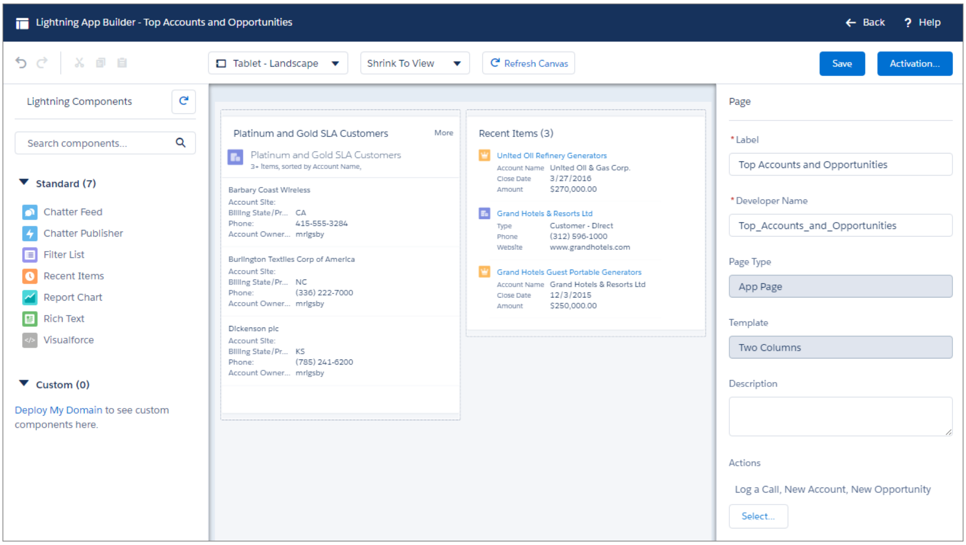
Let's get started!

1. From Setup, enter `App Builder` in the `Quick Find` box, then select **Lightning App Builder**.
2. Click **New**.
3. Select **App Page**, and then click **Next**.
4. Name your Lightning Page *`Top Accounts and Opportunities`*.
5. Select the **Two Columns** template and click **Finish**.

   If the Lightning App Builder walkthrough pops up, dismiss it.

   Although you selected a two-column template, the canvas shows only one column. When you create an app page, the mode defaults to Phone.
6. Drag the Filter List component into the top region.
7. In the properties pane, select the **Platinum and Gold SLA Customers** filter.
8. Add a Recent Items component into the second region.
9. In the properties pane for the Recent Items component, click **Select** and add the Opportunity object to it.
10. In the properties pane, click **Page**, click **Select**, and then add these actions to the page.
    - Log a Call
    - New Account
    - New Opportunity

    You can add only global actions to a Lightning Page.
11. From the toolbar, change the Phone view to Tablet-Landscape.
12. Click **Save**, and then **Not Yet**.



Deploy My Domain to see custom components here.

Platinum and Gold SLA Customers

Platinum and Gold SLA Customers
3+ Items, sorted by Account Name,

Barbary Coast Wireless
Account Site:
Billing State/Pr... CA
Phone: 415-555-3284
Account Owner... mrigsby

Burlington Textiles Corp of America
Account Site:
Billing State/Pr... NC
Phone: (336) 222-7000
Account Owner... mrigsby

Dickenson plc
Account Site:
Billing State/Pr... KS
Phone: (785) 241-6200
Account Owner... mrigsby

Recent Items (3)

United Oil Refinery Generators
Account Name United Oil & Gas Corp.
Close Date 3/27/2016
Amount $270,000.00

Grand Hotels & Resorts Ltd
Type Customer - Direct
Phone (312) 596-1000
Website www.grandhotels.com

Grand Hotels Guest Portable Generators
Account Name Grand Hotels & Resorts Ltd
Close Date 12/3/2015
Amount $250,000.00

Page

* Label: Top Accounts and Opportunities

* Developer Name: Top_Accounts_and_Opportunities

Page Type: App Page

Template: Two Columns

Description:

Actions: Log a Call, New Account, New Opportunity

Now that you've created your page, you're almost ready to add it to Salesforce1 and Lightning Experience.

Add Your App Page to Salesforce1 and Lightning Experience

Just like the other pages, your users can't access your app page until you activate it. During activation, you can customize the page's custom tab label, adjust its visibility, and set its position in the Salesforce1 navigation list and Lightning Experience app navigation bars, all in one place.

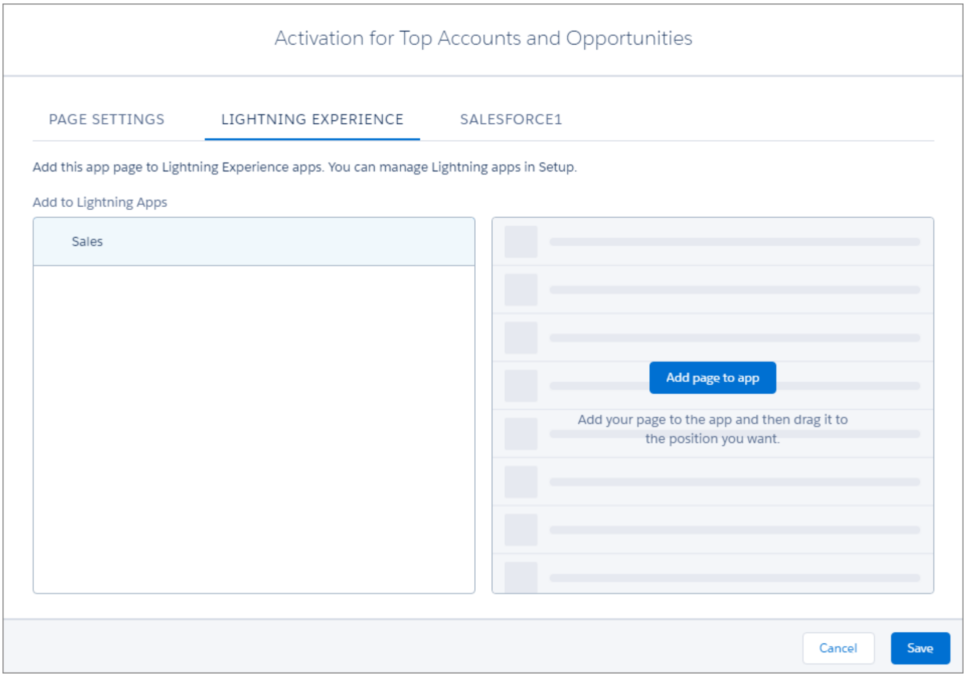
1. Click **Activation**.
2. Don't change the app name.

   By default, the label that you give the Lightning Page is used as the label for its custom tab.
3. Change the icon to the blue lightning bolt icon.

   The icon that you choose here is used as the icon for the app in Salesforce1 and for the page in Lightning Experience.
4. Keep the tab's visibility open to all users.

   Tip: The `Activate for System Administrators only` setting can be useful while you're working on your Lightning Page. Restricting your page to administrators only means that you can see and test the page, but your users can't see it until you're ready to expose it to them.

5. Click the **Lightning Experience** tab.
6. Select a Lightning app, and click **Add page to app**.



Activation for Top Accounts and Opportunities

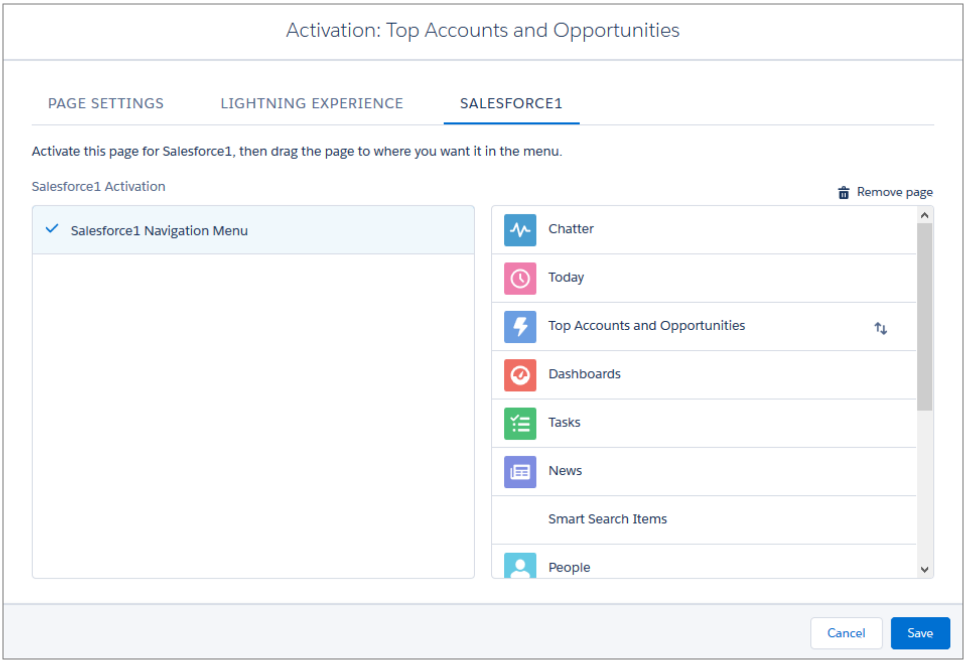
Add this app page to Lightning Experience apps. You can manage Lightning apps in Setup.

Add to Lightning Apps

Add your page to the app and then drag it to the position you want.

The page you're adding to the menu appears in the second position by default. Let's leave it there. If you put it into the top position, it becomes the landing page for all your Lightning Experience users.

7. Click the **Salesforce1** tab.
8. Select the **Salesforce1** Navigation Menu, and click **Add page to app**.
   By default, new pages you add to the Salesforce1 menu appear below the Smart Search Items menu item. If you leave the Top Accounts and Opportunities page there, it will appear in the Apps section of the Salesforce1 menu. We don't want that, so let's move it up.
9. Drag the page to below the Today menu item.



Activation: Top Accounts and Opportunities

PAGE SETTINGS | LIGHTNING EXPERIENCE | SALESFORCE1

Activate this page for Salesforce1, then drag the page to where you want it in the menu.

Salesforce1 Activation

✓ Salesforce1 Navigation Menu

Remove page

- Chatter
- Today
- Top Accounts and Opportunities
- Dashboards
- Tasks
- News
- Smart Search Items
- People

Cancel | Save

10. Click **Save**.

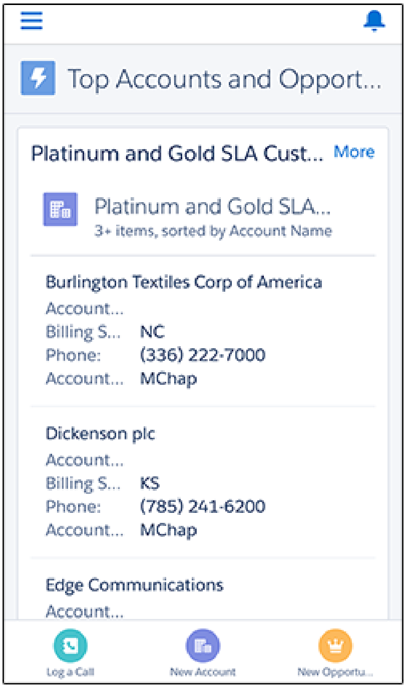
Your app home page is now ready for your Salesforce1 and Lightning Experience users!

Test Your App Page in Salesforce1

You've created your page and activated it. Now let's see it in action!

1. Open the Salesforce1 app on your mobile device.
2. Log in using your Developer Edition credentials.
3. Tap ☰ to access the navigation menu.
4. Select **Top Accounts and Opportunities**.

   Here's your Lightning Page! It has the lightning bolt icon you chose, and the three actions that you added are in the action bar.



Top Accounts and Opport...

Platinum and Gold SLA Cust... More

Platinum and Gold SLA...
3+ items, sorted by Account Name

Burlington Textiles Corp of America
Account...
Billing S... NC
Phone: (336) 222-7000
Account... MChap

Dickenson plc
Account...
Billing S... KS
Phone: (785) 241-6200
Account... MChap

Edge Communications
Account...

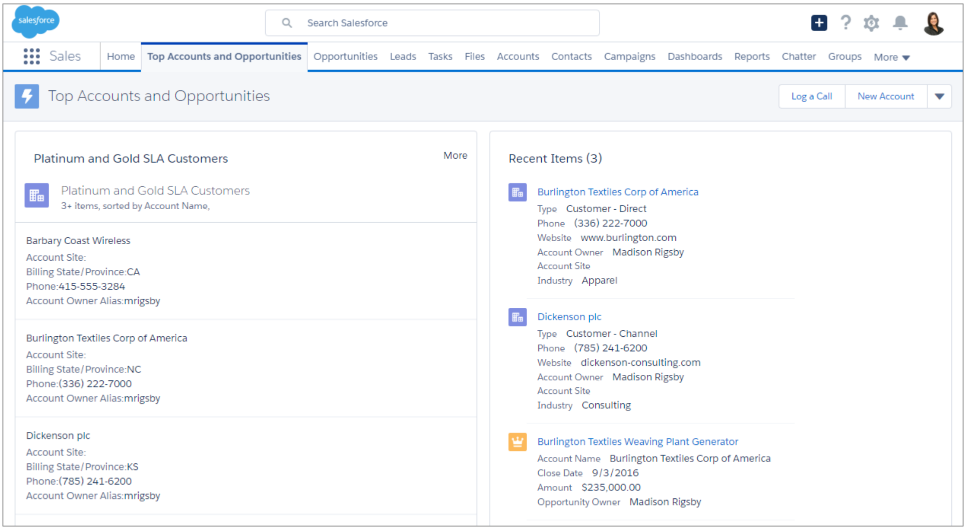
5. Play around with your page. Scroll up and down to see the components, and tap an action icon to see what happens.

Test Your App Page in Lightning Experience

Now let's look at it in Lightning Experience.

1. Go back to your desktop browser and your Developer org.
2. If you're still in the App Builder, click **Back** to return to Setup.
3. From the App Launcher (⋮⋮⋮), open the Lightning app you assigned your page to.
4. Click **Top Accounts and Opportunities** from the app navigation bar.

   Here's your Lightning Page! The lightning bolt icon appears here too, and the three actions that you added are in the highlights panel.



Top Accounts and Opportunities

Platinum and Gold SLA Customers

Platinum and Gold SLA Customers
3+ items, sorted by Account Name,

Barbary Coast Wireless
Account Site:
Billing State/Province:CA
Phone:415-555-3284
Account Owner Alias:mrigsby

Burlington Textiles Corp of America
Account Site:
Billing State/Province:NC
Phone:(336) 222-7000
Account Owner Alias:mrigsby

Dickenson plc
Account Site:
Billing State/Province:KS
Phone:(785) 241-6200
Account Owner Alias:mrigsby

Recent Items (3)

Burlington Textiles Corp of America
Type Customer - Direct
Phone (336) 222-7000
Website www.burlington.com
Account Owner Madison Rigsby
Account Site
Industry Apparel

Dickenson plc
Type Customer - Channel
Phone (785) 241-6200
Website dickenson-consulting.com
Account Owner Madison Rigsby
Account Site
Industry Consulting

Burlington Textiles Weaving Plant Generator
Account Name Burlington Textiles Corp of America
Close Date 9/3/2016
Amount $235,000.00
Opportunity Owner Madison Rigsby

MIGRATE JAVASCRIPT BUTTON FUNCTIONALITY TO LIGHTNING

CHAPTER 18 JavaScript Buttons: It's Time to Move On

In this chapter ...

- JavaScript Button Security Issues and Use Cases

We know that you love JavaScript buttons and have been using them in Salesforce Classic for years. In fact, you might be reluctant to migrate to Lightning Experience because JavaScript buttons aren't supported. But Lightning Experience offers so much more than Salesforce Classic, and it is the future. We realize that you expect and rely on us to always migrate existing functionality to new features and UI, but in the case of JavaScript buttons, we believe that the future is brighter in Lightning Experience, even without JavaScript button support. And in this module, we show you why.

JavaScript buttons and links are types of actions in the Salesforce Classic UI that let you create inline JavaScript code that can be invoked via a button or link embedded on a record or list page. For example, maybe you prepopulate new records with data upon creation and update values in fields based on other logic. Or maybe you're a Salesforce partner who uses custom buttons to integrate with your platform.

If JavaScript buttons are so useful, why don't we support them in Lightning Experience? Because there are significant security challenges with combining untrusted JavaScript from multiple sources and authors together with the application source code, while maintaining trust.

We'll cover those security and functional challenges, and share with you the alternatives to JavaScript buttons that are mobile- and Lightning-friendly. We'll also look at features in Salesforce that you can use to migrate the functionality that you've built using custom buttons.

We're committed to solving the problem of client-side customization and integration. Let us show you a new approach to thinking about JavaScript button functionality in Lightning Experience.

JavaScript Button Security Issues and Use Cases

One of the coolest benefits of Lightning Experience is that you can add your custom Lightning components to record, Home, and other pages. For example, you might choose to add a map component to your Account record pages. Or, you could provide a component for your AppExchange app that can be added to the Home page or an Opportunity record.

However, without some safeguards, the components have access to each other's data, shared access to the window and event structures, and access to any client-side API. A partner's component for HIPAA compliance or financial information, for example, could be accessed by a component from a different source when both components are on the same page. As you can imagine, this cross-component access could lead to security and regulatory issues.

What's Up with Inline JavaScript

Before we discuss the safeguards that Salesforce has in place for Lightning component security, let's highlight some of the issues with in-line JavaScript. JavaScript is a loosely typed programming language, supported by all modern web browsers without a plug-in. It can persist data and state through cookies and storage APIs, and it can access events, URLs, and cookies through the browser. What makes JavaScript both useful and dangerous is that it has full access to the Document Object Model (DOM) and Browser Object Model (BOM).

With access to the DOM, a programmer can add, change, or delete almost anything found in an HTML or XML document. In the right hands, this is useful because JavaScript provides an API for working with text, dates, and regular expressions, so it's easy to add client-side functionality with JavaScript snippets that enhance the base user interface. However, this is also a significant vulnerability because with Cross Site Scripting (XSS), malicious actors can gain access via JavaScript to the DOM or BOM and wreak havoc.

When a website enables dynamic content, hackers can use XSS to inject malicious client-side code into the web pages that are viewed by normal users. The hackers can then harness a user's session and cookies to run scripts to extract data, log keystrokes, manipulate form entries, and even access APIs.

LockerService: Making Lightning Components More Secure

The good news is that Salesforce is already working on a solution to make Lightning components more secure and restrict JavaScript's unfettered access. This solution is LockerService, which uses various technologies and techniques that are intended to prevent:

- XSS and similar security issues
- Unrestricted DOM access

- Calls to undocumented/private APIs

At the same time, LockerService's features also enable:

- Client-side API versioning
- Faster security review (AppExchange)
- Better JavaScript development practices
- Easy updates to security features and policies

So you now know that Lightning components are built to be more secure. But how can you benefit from using them, and how can you re-create your JavaScript button functionality within Lightning Experience? We'll show you shortly. But first, let's look at how you might be using JavaScript buttons in Salesforce Classic.

What People Are Doing with JavaScript Buttons

We heard from many customers, some with hundreds of JavaScript buttons in their orgs. We also talked to partners about their JavaScript button use cases. We collated what we learned into a broader set of operations. Here are the most common use cases for JavaScript buttons.

- Use or manipulate values on a record *before* the save
  - Validate fields—ensure that values are populated and/or meet criteria
  - Prefill values based on inputs in other fields
  - Redirect to a Visualforce page based on input values
  - Display confirmation pop-up screens
- Create records with prepopulated values
- Trigger flows from Visual Workflow
- Call out to Salesforce or external APIs
- Integrate with third parties
- Perform mass actions on records in a list
- Direct methods and procedures via feedback pop-up screens for users

There are more scenarios, and some use cases that are so specific to an org that they're impossible to categorize. Coming up, we'll cover features that you can use to address all the use cases we mentioned, and migrate JavaScript button functionality to Salesforce1 and Lightning Experience.

CHAPTER 19 You Have Buttons? We Have Alternatives

In this chapter ...

- Quick Actions
- Custom URL Buttons and Links
- Apex Triggers
- Custom Visualforce Buttons

Previously, we explained why it's time to migrate to Lightning Experience, where great new features let you move past JavaScript buttons. Next, we'll explain how you can easily migrate your custom JavaScript button functionality using solutions that work in both Salesforce Classic and Lightning Experience.

This table maps use cases for JavaScript buttons to alternate—and in most cases, better—solutions in Lightning.

JavaScript Button Top Use Cases	Lightning Alternatives	Declarative/Programmatic
Validate fields (presave)	Quick actions (using default values and/or formulas)	D
	Apex triggers	P
Create records with prepopulated values	Quick actions (using default values and/or formulas)	D
Redirect to a record page	Custom URL buttons	D
Redirect to a Visualforce page	Visualforce quick actions	P
	Lightning actions	P
Prefill values based on inputs	Lightning actions	P
Confirmation pop-up screens	Lightning actions	P

JavaScript Button Top Use Cases	**Lightning Alternatives**	**Declarative/Programmatic**
API calls (Salesforce and third-party)	Lightning actions	P
Feedback pop-up screens	Lightning actions	P
Third-party integration	Lightning actions	P
Mass actions on list view records	Custom Visualforce buttons on list views	P

As you can see, Salesforce offers several declarative tools for converting the functionality of your JavaScript custom buttons.

Quick Actions

Quick actions support many of the common uses of JavaScript buttons. Quick actions can be based on a specific object or can be global, meaning that they're more generic and accessible from any record or the Chatter feed. There's a quick action to do just about anything in Salesforce. Here are some examples.

Validate Field Values

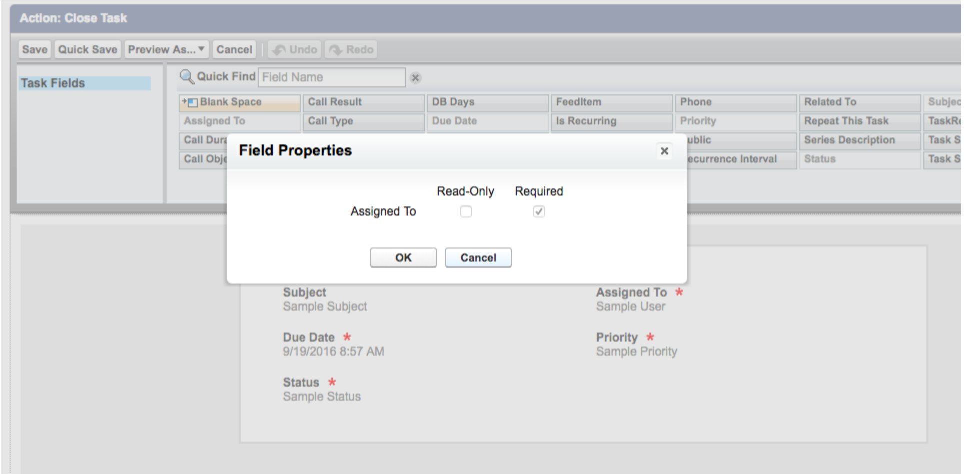
Sometimes you want to make sure that certain fields are filled in or populated with specific criteria when your users create or update records.

Let's say you want to create an action for closing a task without requiring users to go to a full edit page. But you also want to make sure that the task has a due date before it can be closed.

You can fulfill all these requirements by creating a quick action for the Task object.

1. In the object management settings for Activities, go to Task Buttons, Links, and Actions.
2. Click **New Action**.
3. For Action Type, select **Update a Record**.
4. For Label, type in `Close Task`.
5. Click **Save**.

Now we select the fields that we want to appear on the "Close Task" quick action. You can easily make a field required or read-only through its field properties.



	Read-Only	Required
Assigned To	☐	☑

Subject
Sample Subject

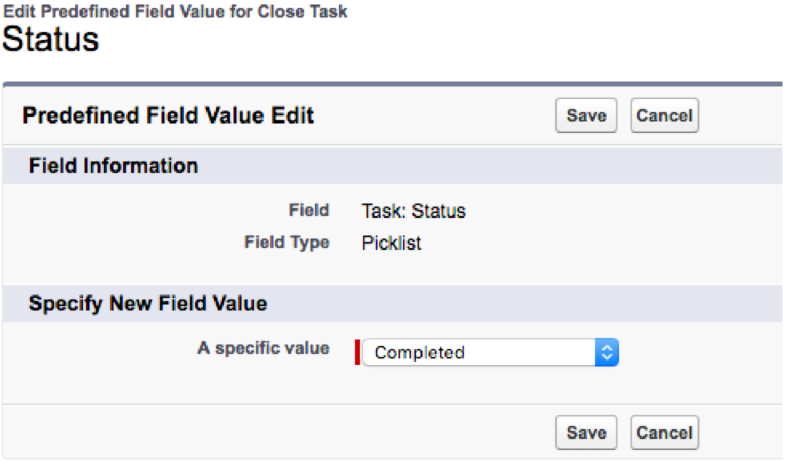
Assigned To *
Sample User

Due Date *
9/19/2016 8:57 AM

Priority *
Sample Priority

Status *
Sample Status

After you set the fields on the action layout, you can add predefined field values for any of the fields on the task record. In this example, we marked the Due Date field as required. We also added a predefined value for the Status field to be changed to Completed.



Edit Predefined Field Value for Close Task

Status

Predefined Field Value Edit

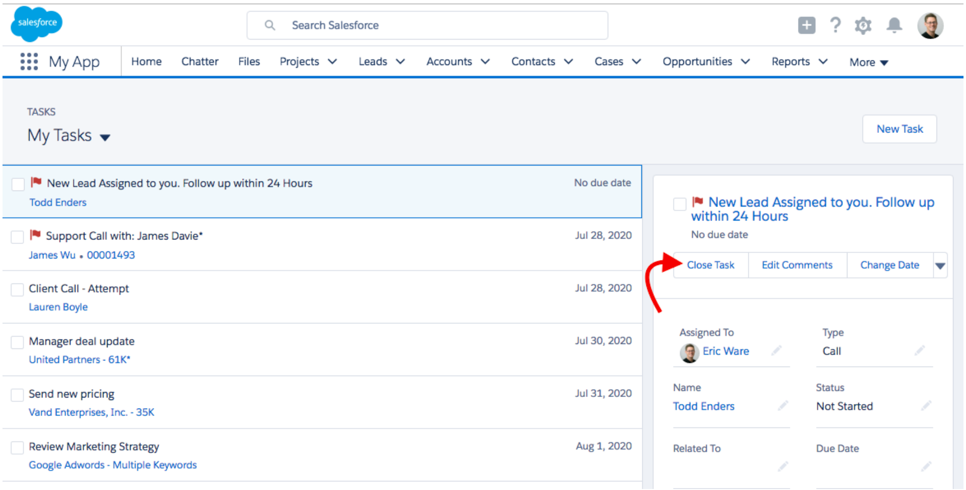
Field Information

Field	Task: Status
Field Type	Picklist

Specify New Field Value

A specific value: Completed

Now that we're done configuring the action, we add it to the Task page layout. Then users can access it from a task record page in both Lightning Experience and Salesforce1. Here's an example of a Close Task action on the Tasks page.



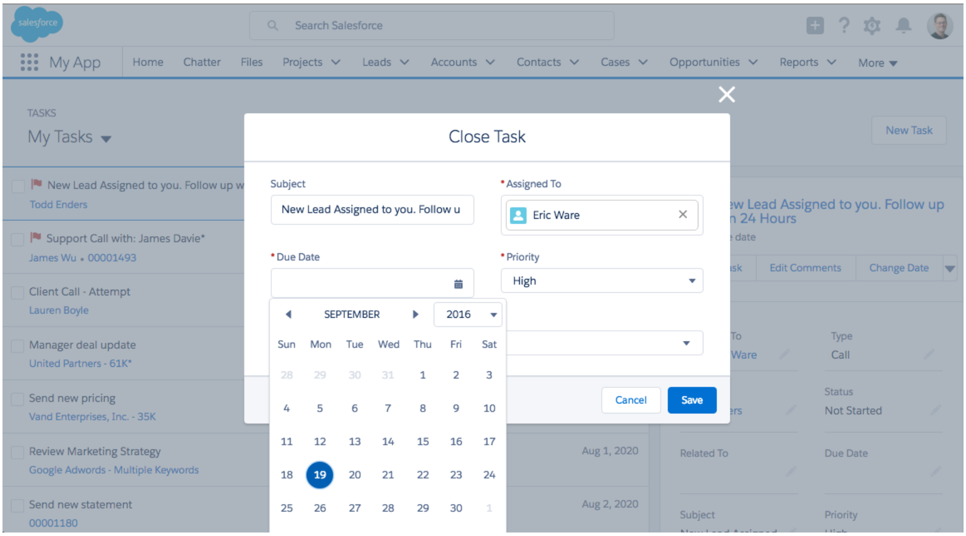
New Lead Assigned to you. Follow up within 24 Hours Todd Enders	No due date
Support Call with: James Davie* James Wu • 00001493	Jul 28, 2020
Client Call - Attempt Lauren Boyle	Jul 28, 2020
Manager deal update United Partners - 61K*	Jul 30, 2020
Send new pricing Vand Enterprises, Inc. - 35K	Jul 31, 2020
Review Marketing Strategy Google Adwords - Multiple Keywords	Aug 1, 2020

New Lead Assigned to you. Follow up within 24 Hours

No due date

Assigned To Eric Ware	Type Call
Name Todd Enders	Status Not Started
Related To	Due Date

Clicking **Close Task** brings up the action, which the user can quickly act upon and save.



Close Task

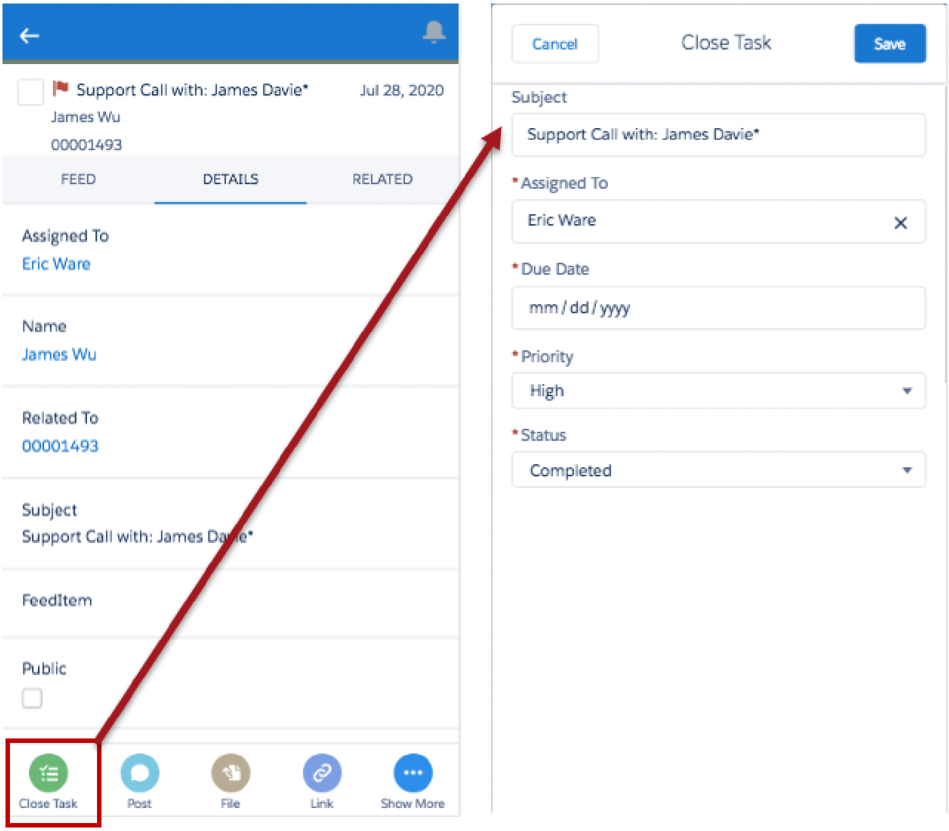
Subject: New Lead Assigned to you. Follow u

*Assigned To: Eric Ware

*Due Date

*Priority: High

Here's what it looks like in Salesforce1.



Support Call with: James Davie* Jul 28, 2020
James Wu
00001493

Assigned To
Eric Ware

Name
James Wu

Related To
00001493

Subject
Support Call with: James Davie*

FeedItem

Public
□

Subject
Support Call with: James Davie*

*Assigned To
Eric Ware

*Due Date
mm / dd / yyyy

*Priority
High

*Status
Completed

Prepopulate Fields with Values

A more advanced use case is when you want to let users create a record, but you also want one or more of the fields to be populated automatically based on values in a dependent field.

For example, let's say your inside sales team's typical quarterly quota is one fourth of their customers' revenues from last year, increased by 10%. Because that simple formula doesn't always apply, you want to prefill the opportunity amount with the value, but also let the user modify it. Create an action so your users can modify the field quickly and efficiently without going to the full opportunity record page.

To create this sample action, you follow the same steps as before for creating the opportunity quick action.

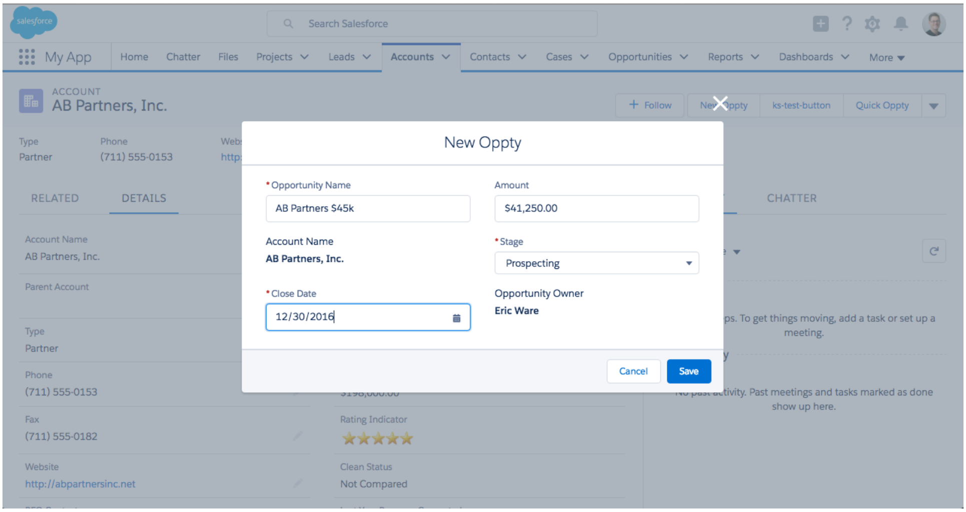
1. In the object management settings for accounts, go to Buttons, Links, and Actions.
2. Click **New Action**.
3. For Action Type, select **Create a Record**.

4. For Target Object, select **Opportunity**.
5. Pick the appropriate Record Type.
6. For Label, enter `New Oppty`.
7. Click **Save**.

Once you've picked the fields for the action layout, you can add predefined values for the Amount field. In our example, we've used this formula:

```
Account.Last_Year_Revenue_Generated__c  * 1.10  / 4
```

Add this action to the Account page layout. When users invoke it, they see a value prepopulated in the field that they can accept or override.



New Oppty

*Opportunity Name: AB Partners $45k

Amount: $41,250.00

Account Name: AB Partners, Inc.

*Stage: Prospecting

*Close Date: 12/30/2016

Opportunity Owner: Eric Ware

The cool thing about this action is that you can pull data from the opportunity's account to use when creating the opportunity. Salesforce also supports record traversal, so if you have hierarchical accounts, you can pull the revenues from the parent account like this:

```
Account.Parent.Last_Year_Revenue_Generated__c
```

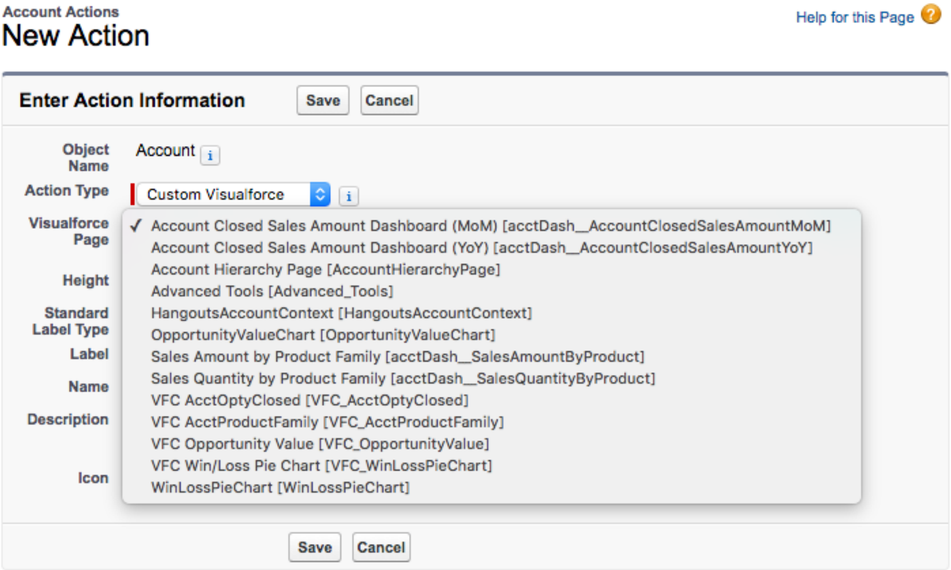
Don't worry about remembering the formula formats; quick actions are declarative and use the Salesforce formula builder.

Redirect to a Visualforce Page Based on Input Values

You can create Visualforce pages to enhance your business processes. Users can navigate to these Visualforce pages in various ways, such as with custom buttons, action overrides, and tabs.

One benefit of Visualforce pages is that by using the standard controller, you can create customized record pages and add prevalidation, prepopulated fields, formulas, and much more.

JavaScript buttons are commonly used in Salesforce Classic to read and pass values from a record into a URL that then redirects users to a Visualforce page. You can also give your users access to Visualforce pages via quick actions. Creating Visualforce quick actions is easy, and a similar process to what we've already covered. The only difference is that you select Custom Visualforce as the action type.


Account Actions
New Action

Enter Action Information

Object Name: Account

Action Type: Custom Visualforce

Help for this Page
Save
Cancel
Visualforce Page
Height
Standard Label Type
Label
Name
Description
Icon
✓ Account Closed Sales Amount Dashboard (MoM) [acctDash__AccountClosedSalesAmountMoM]
Account Closed Sales Amount Dashboard (YoY) [acctDash__AccountClosedSalesAmountYoY]
Account Hierarchy Page [AccountHierarchyPage]
Advanced Tools [Advanced_Tools]
HangoutsAccountContext [HangoutsAccountContext]
OpportunityValueChart [OpportunityValueChart]
Sales Amount by Product Family [acctDash__SalesAmountByProduct]
Sales Quantity by Product Family [acctDash__SalesQuantityByProduct]
VFC AcctOptyClosed [VFC_AcctOptyClosed]
VFC AcctProductFamily [VFC_AcctProductFamily]
VFC Opportunity Value [VFC_OpportunityValue]
VFC Win/Loss Pie Chart [VFC_WinLossPieChart]
WinLossPieChart [WinLossPieChart]
Save
Cancel

For object-specific Visualforce quick actions, you must include the `standardController` for the object in your Visualforce page to gain access to the record data and have the Visualforce page appear in the quick action picklist.

Custom URL Buttons and Links

Maybe you use JavaScript buttons for navigation, redirecting users to another page with the command `window.open(URL)` and some variables. In most cases, you can use custom URL buttons or links in Lightning Experience instead.

Here's a list of different URL buttons and links, and their redirect behavior in Lightning Experience.

Custom URL Button or Link	Lightning Experience Behavior
External URL	URL opens in new tab

Custom URL Button or Link	Lightning Experience Behavior
`www.google.com`	
Relative Salesforce URL, View `/{!Account.Id}`	Record home page opens in existing tab
Relative Salesforce URL, Edit `/{!Account.Id}/e`	Edit overlay pops up on the existing page
Relative Salesforce URL, List `/001/o`	Object home page opens in existing tab

$Action URL, View
`{!URLFOR($Action.Account.View,`
`Account.Id)}`
	Record home page opens in existing tab

$Action URL, Edit
`{!URLFOR($Action.Account.Edit,`
`Account.Id)}`
	Edit overlay pops up on the existing page

Apex Triggers

You might be familiar with Apex triggers; they've been supported on our platform for years. Apex triggers can be configured to execute before or after a user clicks Save on a record.

When you need presave validation, calculation, and population of fields, consider using Apex triggers. They're especially useful for third-party integration, because the rules are enforced through the Salesforce API across Salesforce Classic, Lightning Experience, and Salesforce1.

For more information on Apex triggers, you can check out the *Apex Developer Guide* or earn the Apex Triggers badge.

Custom Visualforce Buttons

Another great feature in Lightning Experience is support for using Visualforce buttons on list views. With this feature, you can use existing Visualforce actions in Lightning and work with multiple records in lists. Here's how.

1. Create your Visualforce page.

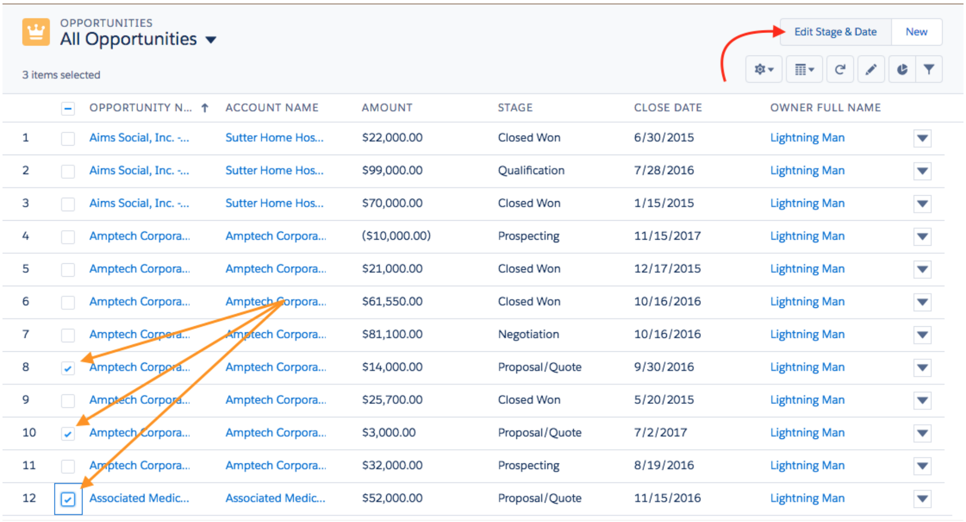
   Here is sample code for editing the Stage and Close dates for multiple opportunities:

   ```
   <apex:page standardController="Opportunity"
   recordSetVar="opportunities" extensions="tenPageSizeExt">
      <apex:form>
         <apex:pageBlock title="Edit Stage and Close Date"
   mode="edit">
            <apex:pageMessages />
            <apex:pageBlockButtons location="top">
               <apex:commandButton value="Save" action="{!save}"/>
               <apex:commandButton value="Cancel"
   action="{!cancel}"/>
            </apex:pageBlockButtons>
            <apex:pageBlockTable value="{!selected}" var="opp">
               <apex:column value="{!opp.name}"/>
               <apex:column headerValue="Stage">
                  <apex:inputField value="{!opp.stageName}"/>
               </apex:column>
               <apex:column headerValue="Close Date">
                  <apex:inputField value="{!opp.closeDate}"/>
               </apex:column>
            </apex:pageBlockTable>
         </apex:pageBlock>
      </apex:form>
   </apex:page>
   ```

2. Create a custom button that references your Visualforce page.
3. Add the action to your list view.

   Note: Mass actions aren't supported on the Recently Viewed records list. They are only available on list views.

Example: Here's how that action shows up in Lightning Experience.



OPPORTUNITIES

All Opportunities ▾

3 items selected

	−	OPPORTUNITY N... ↑	ACCOUNT NAME	AMOUNT	STAGE	CLOSE DATE	OWNER FULL NAME	
1	□	Aims Social, Inc. -...	Sutter Home Hos...	$22,000.00	Closed Won	6/30/2015	Lightning Man	▼
2	□	Aims Social, Inc. -...	Sutter Home Hos...	$99,000.00	Qualification	7/28/2016	Lightning Man	▼
3	□	Aims Social, Inc. -...	Sutter Home Hos...	$70,000.00	Closed Won	1/15/2015	Lightning Man	▼
4	□	Amptech Corpora...	Amptech Corpora...	($10,000.00)	Prospecting	11/15/2017	Lightning Man	▼
5	□	Amptech Corpora...	Amptech Corpora...	$21,000.00	Closed Won	12/17/2015	Lightning Man	▼
6	□	Amptech Corpora...	Amptech Corpora...	$61,550.00	Closed Won	10/16/2016	Lightning Man	▼
7	□	Amptech Corpora...	Amptech Corpora...	$81,100.00	Negotiation	10/16/2016	Lightning Man	▼
8	☑	Amptech Corpora...	Amptech Corpora...	$14,000.00	Proposal/Quote	9/30/2016	Lightning Man	▼
9	□	Amptech Corpora...	Amptech Corpora...	$25,700.00	Closed Won	5/20/2015	Lightning Man	▼
10	☑	Amptech Corpora...	Amptech Corpora...	$3,000.00	Proposal/Quote	7/2/2017	Lightning Man	▼
11	□	Amptech Corpora...	Amptech Corpora...	$32,000.00	Prospecting	8/19/2016	Lightning Man	▼
12	☑	Associated Medic...	Associated Medic...	$52,000.00	Proposal/Quote	11/15/2016	Lightning Man	▼

As you can see from our examples, Salesforce has a great set of features that allow you to migrate your JavaScript button functionality and move to Lightning Experience.

Now, you may have a lot of JavaScript buttons that have accumulated in your org over the years. You might expect that the migration or conversion process will take a long time. But the job might not be as hard as you think. We did an analysis of JavaScript buttons in the internal Salesforce org that all our employees use. We found that many buttons were obsolete or rarely invoked by users. Others were simply duplicates—the same button but on different objects. After going through the list, we discovered that many of the JavaScript buttons could be converted to the solutions we've looked at so far. For the remainder, we could address quite a few with the new Lightning Actions.

You probably noticed Lightning actions in the list of programmatic solutions as a common replacement for many of the JavaScript button use cases. Lightning actions are easy to build, because they're based on the quick action framework. You set them up similar to Visualforce quick actions. We'll cover Lightning actions more in depth next.

CHAPTER 20 Lightning Actions: Smart, Fast, and Mobile

In this chapter ...

- Populate Fields Based on User Input and Provide Feedback Messages to Users
- Integrate Third-Party APIs

We've covered several solutions that work in both Lightning Experience and Salesforce Classic and that are great replacements for JavaScript buttons. But we recognize that those declarative and programmatic solutions don't address every use case. The good news is that there's something that addresses many more. Introducing Lightning actions: quick actions that call Lightning components.

Lightning actions are built on the existing Lightning Component Framework that you may already have adopted. You can easily convert your existing Lightning components to actions and use them in Salesforce1 and Lightning Experience.

You make a Lightning component invokable as an action by adding one of two interfaces to the component: `force:lightningQuickAction` or `force:lightningQuickActionWithoutHeader`. The first interface adds the standard header with Save and Cancel buttons to the Lightning action overlay, and the other doesn't.

Another useful interface for a Lightning action is `force:hasRecordId`, which provides the record context to the component when it's invoked from a record page. If you set up your Lighting action as a global quick action, you don't need the record context. However, if you want to access a record's data or metadata, you must implement `force:hasRecordId`.

Note: If you haven't built Lightning components before, visit the Lightning Development Center, do the Lightning Component Quick Start project, and complete the Lightning Components Basics module.

Let's dive into Lightning actions. Next, we'll talk about some JavaScript button functionality that can be addressed with Lightning actions instead.

- Populating fields based on user input and providing feedback messages during data entry

- Integrating third-party APIs

Populate Fields Based on User Input and Provide Feedback Messages to Users

Let's say you use JavaScript buttons to validate or manipulate data and give your users feedback or instructions when they work with records. Here's an example to show how easily you can validate or manipulate data in a Lightning component before your users create or update a record.

We've built a custom object in our sample org called Case Study, which we use to track different research projects. When we add new test users, we capture their names and email addresses. Since the email address is our primary contact method, we want to make sure it's entered correctly. We also want to create a unique nickname for test users so that they can remain anonymous.

We're going to create a Lightning action that displays the name and email address fields, validates the email address, and uses the first name and a random number to automatically create the nickname.

Here's the code for the component called by this Lightning action.

- `CreateUser.cmp`—The Lightning component that you see when you open the action. It contains the UI implementation, which includes the text fields, buttons, action title, and so on.
- `CreateUserController.js`—The controller that listens to the component and any UI events that take place, such as the initial load, text input, and button clicks. Its function is to delegate these events to the helper class, `CreateUserHelper.js`.
- `CreateUserHelper.js`—The code that deals with all the business logic, such as validating the password and email fields, and communicating with the server-side Apex controller, which saves the data.
- `SaveTestUser.apxc`—A simple Apex controller that handles the request to create a test user.

CreateUser.cmp

```
<aura:component
implements="force:lightningQuickActionWithoutHeader,force:hasRecordId"
 controller="SaveTestUser" >
   <aura:attribute name="user" type="Test_User__c" default="{
'sobjectType': 'Test_User__c' }"/>
  <aura:attribute name="hasErrors" type="Boolean" description="Indicate
 whether there were failures or not" />
   <aura:attribute name="caseStudy" type="String" />
   <aura:attribute name="recordId" type="String"/>

   <aura:handler name="init" value="{!this}" action="{!c.init}" />
   <div class="slds-page-header" role="banner">
      <p class="slds-text-heading--label">Case Study</p>
      <h1 class="slds-page-header__title slds-m-right--small
slds-truncate slds-align-left" title="Case Study
Title">{!v.caseStudy}</h1>
   </div>
   <br/>

   <aura:if isTrue="{!v.hasErrors}">
      <!-- Load error -->
      <div class="userCreateError">
         <ui:message title="Error" severity="error" closable="true">
            Please review the error messages.
         </ui:message>
      </div>
   </aura:if>

   <div class="slds-form--stacked">

      <div class="slds-form-element">
        <label class="slds-form-element__label" for="firstName">Enter
 first name: </label>
         <div class="slds-form-element__control">
            <ui:inputText class="slds-input" aura:id="firstName"
value="{!v.user.first}" required="true" keydown="{!c.updateNickname}"
 updateOn="keydown"/>
         </div>
      </div>

      <div class="slds-form-element">
         <label class="slds-form-element__label" for="lastName">Enter
 last name: </label>
         <div class="slds-form-element__control">
            <ui:inputText class="slds-input" aura:id="lastName"
value="{!v.user.last}" required="true" />
         </div>
      </div>

      <div class="slds-form-element">
         <label class="slds-form-element__label" for="nickname">Enter
 nickname: </label>
         <div class="slds-form-element__control">
            <ui:inputText class="slds-input" aura:id="nickname"
```

```
value="{!v.user.nickname}" required="false"/>
         </div>
      </div>

      <div class="slds-form-element">
        <label class="slds-form-element__label" for="userEmail">Enter
 user's email:</label>
         <div class="slds-form-element__control">
            <ui:inputEmail class="slds-input" aura:id="userEmail"
value="{!v.user.Email__c}" required="true"/>
         </div>
      </div>

      <div class="slds-form-element">
         <label class="slds-form-element__label"
for="userPassword">Enter user's password:</label>
         <div class="slds-form-element__control">
            <ui:inputSecret class="slds-input" aura:id="userPassword"
 value="{!v.user.Password__c}" required="true"/>
         </div>
      </div>

      <div class="slds-form-element">
         <ui:button class="slds-button slds-button--neutral"
press="{!c.cancel}" label="Cancel" />
         <ui:button class="slds-button slds-button--brand"
press="{!c.saveUserForm}" label="Save User" />
      </div>
   </div>

</aura:component>
```

CreateUserController.js

```
({
   init : function(component, event, helper) {
      var action = component.get("c.getCaseStudy");
      action.setParams({"recordId": component.get("v.recordId")});

      action.setCallback(this, function(response) {
         var state = response.getState();
         if(component.isValid() && state == "SUCCESS"){
             component.set("v.caseStudy", response.getReturnValue());

          } else {
             console.log('There was a problem and the state is:
'+state);
          }
      });
      $A.enqueueAction(action);
   },

   updateNickname: function(component, event, helper) {
      // Update the nickname field when 'tab' is pressed
      if (event.getParams().keyCode == 9) {
         var nameInput = component.find("firstName");
         var nameValue = nameInput.get("v.value");
         var nickName = component.find("nickname");
         var today = new Date();
         nickName.set("v.value", nameValue + today.valueOf(today));

      }
   },

   saveUserForm : function(component, event, helper) {
      var name = component.get("v.user.first");
      var last = component.get("v.user.last");
      var password = component.get("v.user.Password__c");
      var email = component.get("v.user.Email__c");
      var nickname = component.get("v.user.nickname");

      var passwordCmp = component.find("userPassword");
      var emailCmp = component.find("userEmail");

      helper.validatePassword(component, event, helper);
      helper.validateEmail(component, event, helper);

      if (passwordCmp.get("v.errors") == null &&
emailCmp.get("v.errors") == null) {
         component.set("v.hasErrors", false);
         helper.save(component,name + " " +
last,password,email,nickname);
      } else {
         component.set("v.hasErrors", true);
      }
   },
   cancel : function(component, event, helper) {
      $A.get("e.force:closeQuickAction").fire();
   }
})
```

CreateUserHelper.js

```
({
   save: function(component, name, password, email, nickname) {
      //Save the user and close the panel
      var action = component.get("c.saveUser");
         action.setParams({
            "name": name,
            "password": password,
            "email": email,
            "nickname": nickname,
            "caseStudy": component.get("v.recordId")
         });

      action.setCallback(this, function(a) {
         var response = a.getReturnValue();
         var state = action.getState();
         if(component.isValid() && state == "SUCCESS"){
            $A.get("e.force:closeQuickAction").fire();
            var toastEvent = $A.get("e.force:showToast");
            toastEvent.setParams({
                  "title": "Success!",
               "message": "The test user has been created."
            });
            toastEvent.fire();
            $A.get('e.force:refreshView').fire();
         } else if (state == "ERROR") {
            console.log('There was a problem and the state is: '+
action.getState());
         }
      });
      $A.enqueueAction(action);
      },

   validatePassword : function(component, event, helper) {
      var inputCmp = component.find("userPassword");
      var value = inputCmp.get("v.value");

      if (value == undefined) {
         inputCmp.set("v.errors", [{message: "You must enter a
password."}]);
      } else if (value.length < 7 || value.length > 15) {
        inputCmp.set("v.errors", [{message: "The password is the wrong
 length: " + value}]);
      } else if (value.search(/[0-9]+/) == -1) {
        inputCmp.set("v.errors", [{message: "The password must contain
 at least one number: " + value}]);
      } else if (value.search(/[a-zA-Z]+/) == -1) {
        inputCmp.set("v.errors", [{message: "The password must contain
 at least one letter: " + value}]);
      } else {
         inputCmp.set("v.errors", null);
      }
      },

   validateEmail : function(component, event, helper) {
      var inputCmp = component.find("userEmail");
      var value = inputCmp.get("v.value");

      if (value == undefined) {
         inputCmp.set("v.errors", [{message: "You must enter an
email."}]);
         return;
      }

      var apos = value.indexOf("@");
      var dotpos = value.lastIndexOf(".");

      if (apos<1||dotpos-apos<2){
         inputCmp.set("v.errors", [{message: "Email is not in the
correct format: " + value}]);
      } else if (value.substring(apos+1, dotpos) != "gmail") {
         inputCmp.set("v.errors", [{message: "Email must be a gmail
account: " + value.substring(apos+1, dotpos)}]);
      } else {
         inputCmp.set("v.errors", null);
      }
      }

})
```

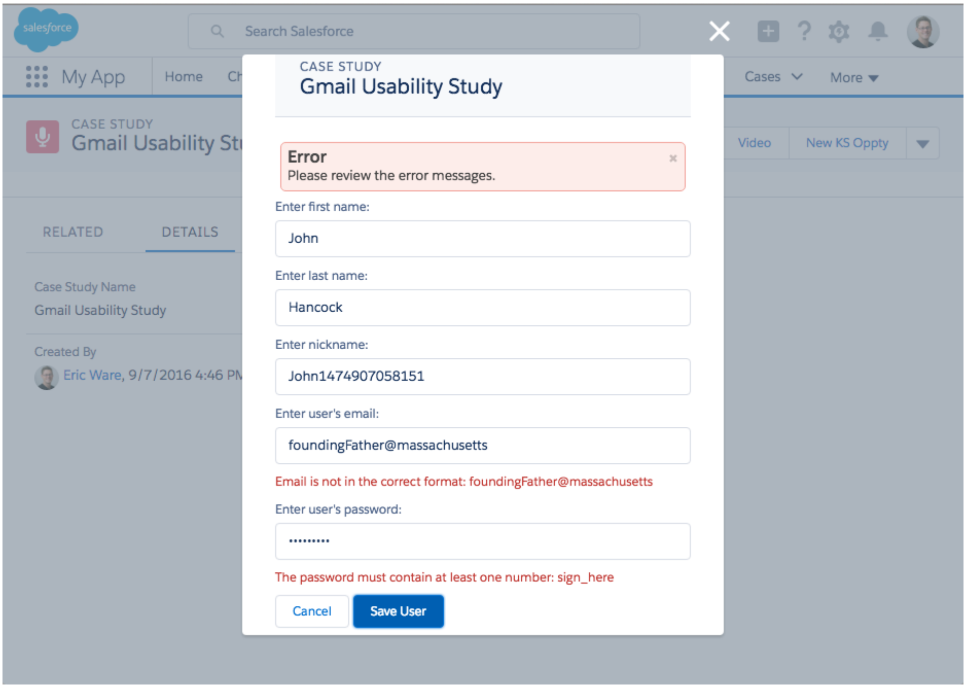
SaveTestUser.apxc

```
public class SaveTestUser {

    @AuraEnabled
    public static Test_User__c saveUser(String name, String password,
 String email, String caseStudy, String nickname) {

        Test_User__c testUser = new Test_User__c(Name=name,
Password__c=password, Email__c=email, Nickname__c=nickname,
Case_Study__c=caseStudy);

        upsert testUser;
        return testUser;
    }

    @AuraEnabled
    public static String getCaseStudy(String recordId) {
        Case_Study__c caseStudyInstance = [SELECT Name FROM
Case_Study__c WHERE id=:recordId];
        return caseStudyInstance.Name;
    }

}
```

After creating the Lightning component, we assign it to an action. In the object management settings for Case Study, we go to Buttons, Links, and Actions, click **New Action**, then configure the action with these parameters.

Field	Value
Object Name	Case Study
Action Type	Lightning Component
Lightning Component	c:CreateUser
Height	500px
Label	Create Test User
Name	CreateUser

Then we add our new Lightning action to the Case Study page layout. When users invoke it from a case study record page, they see the Lightning action that we created.



CASE STUDY
Gmail Usability Study

Error
Please review the error messages.

Enter first name: John

Enter last name: Hancock

Enter nickname: John1474907058151

Enter user's email: foundingFather@massachusetts

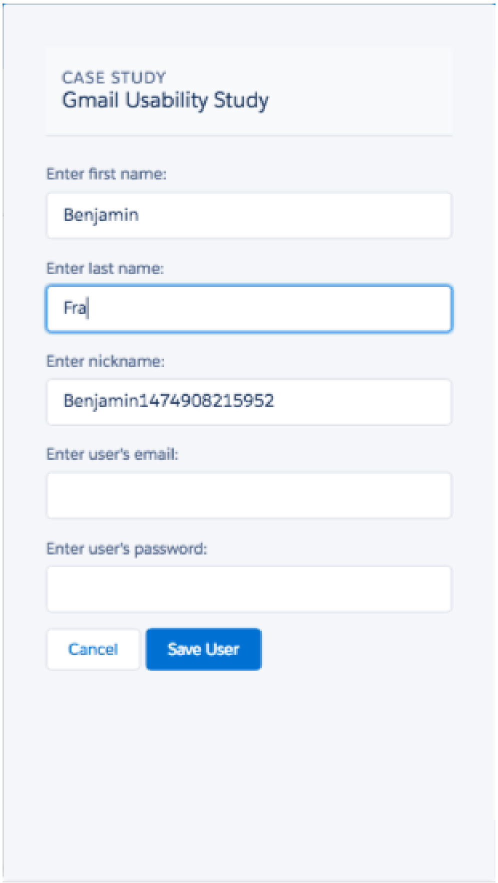
Email is not in the correct format: foundingFather@massachusetts

Enter user's password: •••••••••

The password must contain at least one number: sign_here

Cancel | Save User

The great thing about this Lightning action is that it also works in Salesforce1.



CASE STUDY
Gmail Usability Study

Enter first name:
Benjamin

Enter last name:
Fra

Enter nickname:
Benjamin1474908215952

Enter user's email:

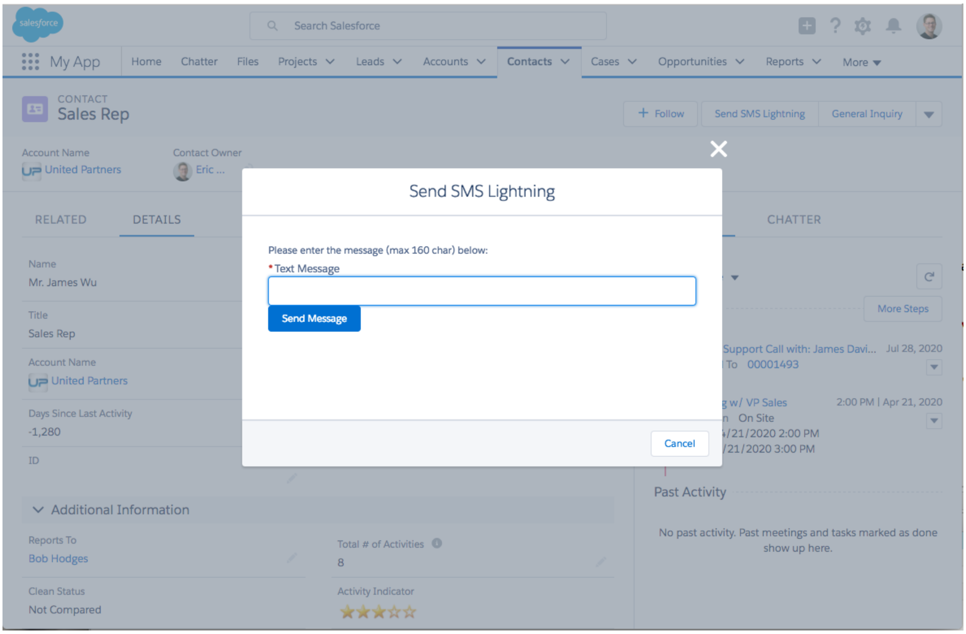
Enter user's password:

Integrate Third-Party APIs

Maybe you use JavaScript buttons for integration with third-party systems. Can you use Lightning actions for that? You sure can. The primary difference is that for integration using Lightning actions, you have to use a server-side controller. In return, you get better handling of security credentials and the ability to use Apex for asynchronous and batch API calls.

Let's look at how to integrate with Twilio for sending SMS messages. In this example, we're in the luxury travel business, and we handle celebrities and VIPs. We want our customer service agents to be able to communicate with their clients, but we don't want to expose the clients' personal contact information to anyone. We create a Lightning action on the Contact object so that agents can send messages without seeing the contact's phone number.

As in our case study example, we'll create the Lightning component and helper classes, and then create a quick action to invoke the component.



This Lightning action is composed of the Lightning component, a JavaScript controller, and an Apex controller, which references library classes that handle the Twilio integration.

SendTwilioSMS.cmp

```
<aura:component controller="TwilioSendSMSController"
implements="flexipage:availableForAllPageTypes,force:hasRecordId,force:lightningQuickAction"
 >
   <aura:attribute name="textMessage" type="String" />
   <aura:attribute name="destinationNumber" type="String" />
   <aura:attribute name="messageError" type="Boolean" />
   <aura:attribute name="recordId" type="String" />

      <aura:handler name="init" value="{!this}" action="{!c.init}" />


   <aura:if isTrue="{!v.messageError}">
      <!-- Load error -->
      <div class="userCreateError">
```

```
         <ui:message title="Error" severity="error" closable="true">
            Unable to send message. Please review your data and try
again.
         </ui:message>
      </div>
   </aura:if>

   <div class="slds-form--stacked">
     <label class="slds-form-element__label" for="instructMsg">Please
 enter the message (max 160 char) below: </label>
      <br/>
      <div class="slds-form-element__control">
        <ui:inputText class="slds-input" aura:id="message" label="Text
 Message" value="{!v.textMessage}" required="true" maxlength="160"
size="165" />
      </div>
      <div class="centered">
         <ui:button class="slds-button slds-button--brand"
press="{!c.sendMessage}" label="Send Message"/>
      </div>
   </div>
</aura:component>
```

SendTwilioSmsController.js

```
({
   init : function(component, event, helper) {
      var action = component.get("c.getPhoneNumber");
      action.setParams({"contactId": component.get("v.recordId")});
      action.setCallback(this, function(response) {
         var state = response.getState();
         if(component.isValid() && state == "SUCCESS"){
            component.set("v.destinationNumber",
response.getReturnValue());
         } else {
            component.set("v.messageError", true);
         }
      });
      $A.enqueueAction(action);
   },

      sendMessage : function(component, event, helper) {
```

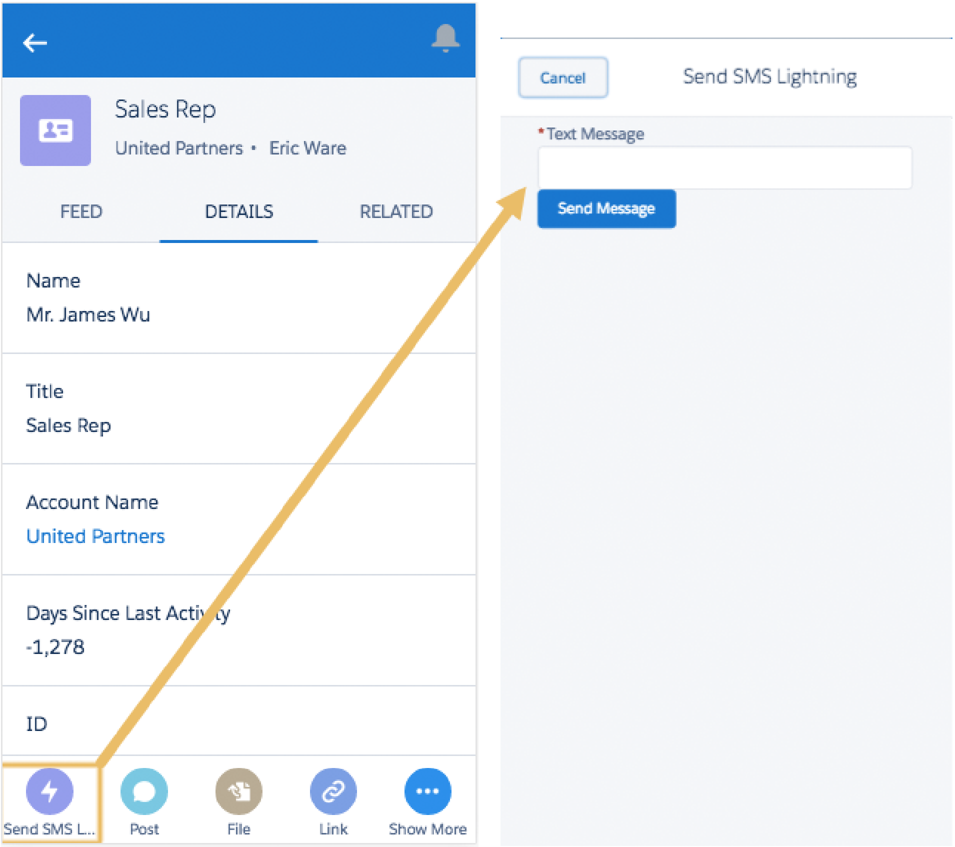
```
      var smsMessage = component.get("v.textMessage");
      var number = component.get("v.destinationNumber");
      var recordId = component.get("v.recordId")

      var action = component.get("c.sendMessages");
      action.setParams({"mobNumber": number, "message": smsMessage,
"contactId": component.get("v.recordId")});
      action.setCallback(this, function(response) {
         var state = response.getState();
         if(component.isValid() && state == "SUCCESS"){
            $A.get("e.force:closeQuickAction").fire();
            var toastEvent = $A.get("e.force:showToast");
            toastEvent.setParams({
               "title": "Success!",
               "message": "SMS has been sent woo hoo!"
            });
            toastEvent.fire();
         } else {
            component.set("v.messageError", true);
         }
      });
      $A.enqueueAction(action);
   }
})
```

SendTwilioSmsController.apxc

```
/*
* Apex controller that currently contains only one method to send sms
 message
*/
global class TwilioSendSMSController {

   /*
   * This method uses the Twilio for Salesforce library class and
method to
   * send the message using the Twilio api.
   */
   @AuraEnabled
      webService static String sendMessages(String mobNumber, String
message, Id contactId) {
         System.debug('the mobNumber is: '+ mobNumber + ' and the
message is: '+ message + ' and contactId is: ' + contactId);

         if (mobNumber == null) {
            mobNumber = getPhoneNumber(contactId);
         }

         try {
            TwilioRestClient client = TwilioAPI.getDefaultClient();

            Map<String,String> params = new Map<String,String> {
               'To' => mobNumber,
               'From' => '15555551234',
               'Body' => message
               };
            TwilioSMS sms =
client.getAccount().getSMSMessages().create(params);
            return sms.getStatus();
         } catch(exception ex) {
            System.debug('oh no, it failed: '+ex);
            return 'failed';
         }
      }

      @AuraEnabled
      public static String getPhoneNumber(Id contactId) {
         Contact currentRecord = [SELECT Phone FROM Contact WHERE Id
= :contactId];
         return currentRecord.Phone.replace(' ', '').replace('-',
'').replace(')', '').replace('(', '');
   }
}
```

After creating the component, we create a Lightning action. This time we add it to the Contact page layout so agents can access it.

Again, the cool thing is that the action is also available in Salesforce1. If a car service provider is trying to contact the customer at the airport, the driver can use a mobile device to reach the customer easily.



Sales Rep

United Partners • Eric Ware

Name

Mr. James Wu

Title

Sales Rep

Account Name

United Partners

Days Since Last Activity

-1,278

ID

Lightning actions are the future of programmatic actions in Lightning Experience. We hope that you start looking at this set of solutions as a better alternative to JavaScript buttons.

And if you depend on partner apps that use JavaScript buttons, you'll be glad to know that many of our partners have already started migrating and upgrading their apps to Lightning. You can find more apps on the AppExchange that are updated for Lightning Experience.

Beyond the Basics

If you'd like to try Twilio for yourself, you can find an unmanaged package of the Twilio for Salesforce library class used in the third-party integration example in the Salesforce Github library: https://github.com/twilio/twilio-salesforce

Before using Twilio for Salesforce and the Twilio API, create a Twilio account, set up a phone number, and connect your account to Salesforce. See the `TwilioApi.cls` in the Github library for more detailed instructions.

ROLL OUT LIGHTNING EXPERIENCE

CHAPTER 21 Welcome to Your Lightning Experience Rollout

In this chapter ...

- What It Takes to Go Live
- Educate Yourself About Lightning Experience
- Getting Hands-On with Lightning Experience
- Identify Stakeholders and an Executive Sponsor
- Educate Your Company About Lightning Experience
- How to Demonstrate Lightning Experience
- Revisit Your Processes
- Perform a Gap Analysis
- Present Your Findings
- When the Show Is a Go

Welcome! Lightning Experience is a new user experience designed to help your sales reps sell faster and smarter. With the launch of Lightning Experience, we've focused on your sales reps with a re-envisioned desktop experience to support your sales process. The result is a more productive interface designed to support how sales reps work on a daily basis.

Although we've started with sales, Lightning Experience won't end there. It's really just the beginning! Lightning Experience will transform Salesforce CRM and extend to service, apps, platform, and more.

Lightning Experience is available for new and existing Salesforce customers. Whether you're a new customer, or you're an existing customer still evaluating the features and the level of effort to enable Lightning Experience for users, this module will help you plan your rollout strategy from start to finish. And developing a strategy is important for any project you undertake with Salesforce. Paying close attention to how your users will transition to the new experience will increase adoption and boost success.

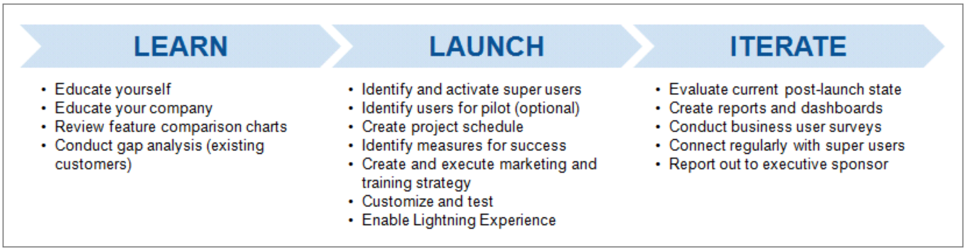
Make no mistake, you play a critical role in this process. As a Salesforce administrator, you are more than just a builder of reports, a creator of fields, or a resetter of passwords. You are, in fact, your company's trusted advisor for Salesforce. And when it comes to Lightning Experience, your company's rollout strategy starts with you.

So let's get this show on the road!

What It Takes to Go Live

Rolling out a project is a lot like putting on a great show. You have to plan it, market it, communicate about it, and execute it. And then the reviews come in. You want rave reviews, which is why you sometimes do previews to work out the kinks before you take your show to the big stage.

As the Salesforce admin, you're the Director of the show. You make it awesome. For the show we're organizing right now ("Lightning Experience Rollout: The Musical," anyone?), you'll plan your rollout, find an executive sponsor, develop and execute a communication strategy, and put on the show. You'll decide whether to run a pilot for a set of users (like a preview) or go big and enable for all users at once (head straight to Broadway).



LEARN

- Educate yourself
- Educate your company
- Review feature comparison charts
- Conduct gap analysis (existing customers)

LAUNCH

- Identify and activate super users
- Identify users for pilot (optional)
- Create project schedule
- Identify measures for success
- Create and execute marketing and training strategy
- Customize and test
- Enable Lightning Experience

ITERATE

- Evaluate current post-launch state
- Create reports and dashboards
- Conduct business user surveys
- Connect regularly with super users
- Report out to executive sponsor

Depending on the size of your organization, you might have a Project Manager assigned to the rollout, or you may be working with a Change Management department at your company. Or you could be charged with organizing and executing the rollout from start to finish. Regardless, your role in getting Lightning Experience live is critical, and it starts with learning everything you can about Lightning Experience.

Let's begin there.

Educate Yourself About Lightning Experience

As you start your strategy for rolling out Lightning Experience, the first thing to do is learn about Lightning Experience. It's easy to do that right here. Complete these three Trailhead modules to get details on the new experience:

- Lightning Experience Basics
- Lightning Experience Features
- App Customization Lite

Already got the badges, but want a quick reminder of the top features? Here's a top features chart for Lightning Experience.

Feature	Why It's Cool
Lightning Apps	Lightning Experience navigation makes your users more efficient by allowing them to complete actions and access recent records and lists directly from the navigation bar. Your users can switch between apps that you can brand with a custom logo, color, and description.
Home	Start the day on the new, intelligent home page. Monitor performance to goal, and get news on key accounts. And stay on track by seeing exactly what needs doing in the Assistant.
Opportunity Workspace	Drive the right behaviors at every stage of your sales process from this action-optimized workspace. You can customize coaching scripts for each step. And reps can work their deals more efficiently using the handy composer to quickly log calls, create tasks, send emails, and more.
Kanban	Visually review the status of leads, opportunities, contracts, and campaigns. Users can drag records from one status to another. For opportunities, user can get alerts when an action is needed on a key deal.
Composer Windows	Breeze through those minor but important activities that inevitably crop up during the day without switching context. Get a call from a customer while reviewing a report? It's no problem to quickly log the call—without needing to search for the related deal or switch to a different record. Composer windows open on the current page and let users log calls and create events, tasks, and notes that can be related to any record.
Console Apps	Empower your users with Lightning console apps. Your users can use Lightning console apps to view and edit multiple records at once, and quickly access utilities like Lightning Voice and Open CTI. Lightning console apps have much of the functionality of the console in Salesforce Classic, such as a three-column record layout and pinned tabs, all backed by the power and flexibility of the Lightning platform.
Activity Timeline and Task List	See open tasks, planned meetings, and accomplishments in the activity timeline on each opportunity, lead, account, and contact. Or use the task list to see the details of specific tasks alongside the full list of open tasks, tasks due today, and closed tasks. Managers can use the Delegated view to see tasks assigned to people below them in the role hierarchy.
Accounts and Contacts	Find information and see news at a glance from the new layout. Get the latest information about customers with integrated Twitter and news, review

Feature	Why It’s Cool
	upcoming and completed activities, and keep data clean with field-level duplicate matching.
Enhanced Notes	Take better notes with auto-save, rich text capabilities, inline images, and versioning. You can even relate a note to multiple records and share notes with teammates or Chatter groups.
Lightning Voice	Use phone features without ever leaving Salesforce. You can make and receive calls, add call notes, and log call information with ease.
Integrated Email	Send email through Gmail or Office 365 accounts with your Salesforce email. See the emails you’ve sent in your Gmail or Office 365 Sent Items folder for seamless integration. With Enhanced Email, you can relate emails to multiple contacts, leads, and users, and to a single opportunity, campaign, case, account, or person account.
File Preview Player	Enjoy a richer file preview experience that doesn’t require Adobe Flash Player. Vector-based preview images render in higher quality and don’t degrade on high-resolution screens, plus they load faster. New controls allow scrolling through multi-page documents, previewing animated GIFs, giving full-screen presentations, and accessing file actions.
Reports and Dashboards	Get more from your data. The new report run page is easier to read and filter. Animated, interactive charts bring your organization’s data to life. And a new dashboard editor supports flexible layouts and components that span columns and rows.
List Views	Seamlessly create and edit custom list views to rapidly find relevant data. See data faster with more intuitive list views, and search on-the-fly for a specific list view. Visualize data with handy list view charts, or apply filters to slice the data as needed.
Search	Find records faster with improved search, which includes recent records and top results.
Setup Tree and Object Manager	Navigate setup pages with a simplified organization and a cleaner look and feel. Use the new Object Manager, which combines all standard and custom objects into a single list that is more easily searchable.
Einstein	Salesforce Einstein brings artificial intelligence into the Salesforce platform. It delivers advanced AI capabilities to sales, service, and marketing—and enables

Feature	Why It's Cool
	anyone to use clicks or code to build AI-powered apps that get smarter with every interaction.

Getting Hands-On with Lightning Experience

One of the best ways to learn about new features is by getting hands-on experience with them. If you'd like to try out Lightning Experience before you enable it in your production org, you've got options.

- Use the Lightning Experience Migration Assistant Preview. The Preview feature lets you explore your production org in Lightning Experience without actually enabling the new interface. You can see exactly how your real data and your current customizations work. You can even change data, settings, permissions—you name it. But remember that because you're working in your live org, changes you make are for real and can impact your users back in Salesforce Classic.
- If you have a sandbox, that's the ideal place to enable the new interface and play around with settings and customizations in earnest. For steps to enable Lightning Experience, see How to Enable Lightning Experience.
- Don't have a sandbox? Get a free Developer Edition or Admin Playground and enable Lightning Experience to give it a test drive.

Once you've got a handle on Lightning Experience, it's time to share it with your company. But who should you share it with?

Identify Stakeholders and an Executive Sponsor

It takes a team to roll out any Salesforce project, and there's no better time to involve your team than right at the beginning. Identify key stakeholders at your company from across all affected departments and form a steering committee, helmed by an executive sponsor who is invested in Salesforce.

Executive Sponsor

Sales Manager

Sales Operations

Salesforce Admin

Super User

This step is critical. Whether you've already committed to rolling out Lightning Experience or you're still evaluating the features, you need executive support to ensure you've got resources and alignment to succeed. In addition, you need the right people from across the company to join your project team, to ensure that every department has their needs met and increase user adoption once you go live.

As you form your project team, keep in mind that depending on the size of your company, you might have people who fill multiple roles (or "wear multiple hats").

Find an executive sponsor and stakeholders, and involve them in your project early and often. Once you identify them, share with them everything you've learned about Lightning Experience.

Educate Your Company About Lightning Experience

One of the most important questions your executive sponsor and stakeholders will have is, "How will Lightning Experience help my team sell more?" Start by showcasing the benefits of Lightning Experience. To help you share Lightning Experience with your company, get the presentation deck included in this enablement pack, which highlights features and benefits. Share the presentation with your stakeholders to help them learn about Lightning Experience.

If you're an existing Salesforce customer, explain what isn't supported in Lightning Experience today to help your company make an educated decision on when to enable the new interface for your different groups of users. To make it easy for you, the presentation also has an appendix with helpful comparison charts.

In addition to sharing the presentation, consider providing a demonstration for your team.

How to Demonstrate Lightning Experience

As your company's trusted advisor, one of the best ways to educate your company on Lightning Experience is to demo it.

Let's go over a few best practices:

Make sure what you demo has data	Make sure you've got sample records for all the features you demo.
Tell your user's stories when you showcase the app	Use your sales team's real life examples as the stories to drive your demo. For example, consider adding their photos to the user profiles.
Test everything beforehand	If you're planning to show any customizations you've added, make sure to test them first.
Practice makes perfect	Run through your demo in advance to make sure you're ready, and to practice the flow of how you'll demo features.
Record your demo!	If you record your demo, you can easily share it with people who couldn't attend in person. The recording can also be a great training asset later!

Revisit Your Processes

When your stakeholders see a demo of Lightning Experience, there will likely be a lot of questions that sound like this: "That's cool, but can it do this? What about that—can it do that?"

These types of questions mean that your users are thinking about how they can take their process and make it work in the new user experience. This is good, but sometimes revisiting a process and improving it is a better next step. In other words, maybe "the way we've always done it" isn't the best path forward. As trusted advisor, you can help your users see that changing their process to adopt new features can actually make them more productive.

So how do you do that? It's easier said than done, but here are some ideas:

- Ask directly, "Can we change the way we do it?" The value proposition of new, improved functionality might be worth updating existing process.
- Share the presentation we provided, which highlights some of the benefits.
- Walk key members of your sales teams and super users through the features, showing them how easy they are, how they work, and why they're helpful.
- Above all, work with your executive sponsor to drive the right behaviors from the top down.

After you demo Lightning Experience, it is time for existing customers to highlight any gaps. This is a critical step to help your company understand what isn't currently supported. The presentation we provided earlier features comparison charts, but you should complete a full gap analysis to share with stakeholders.

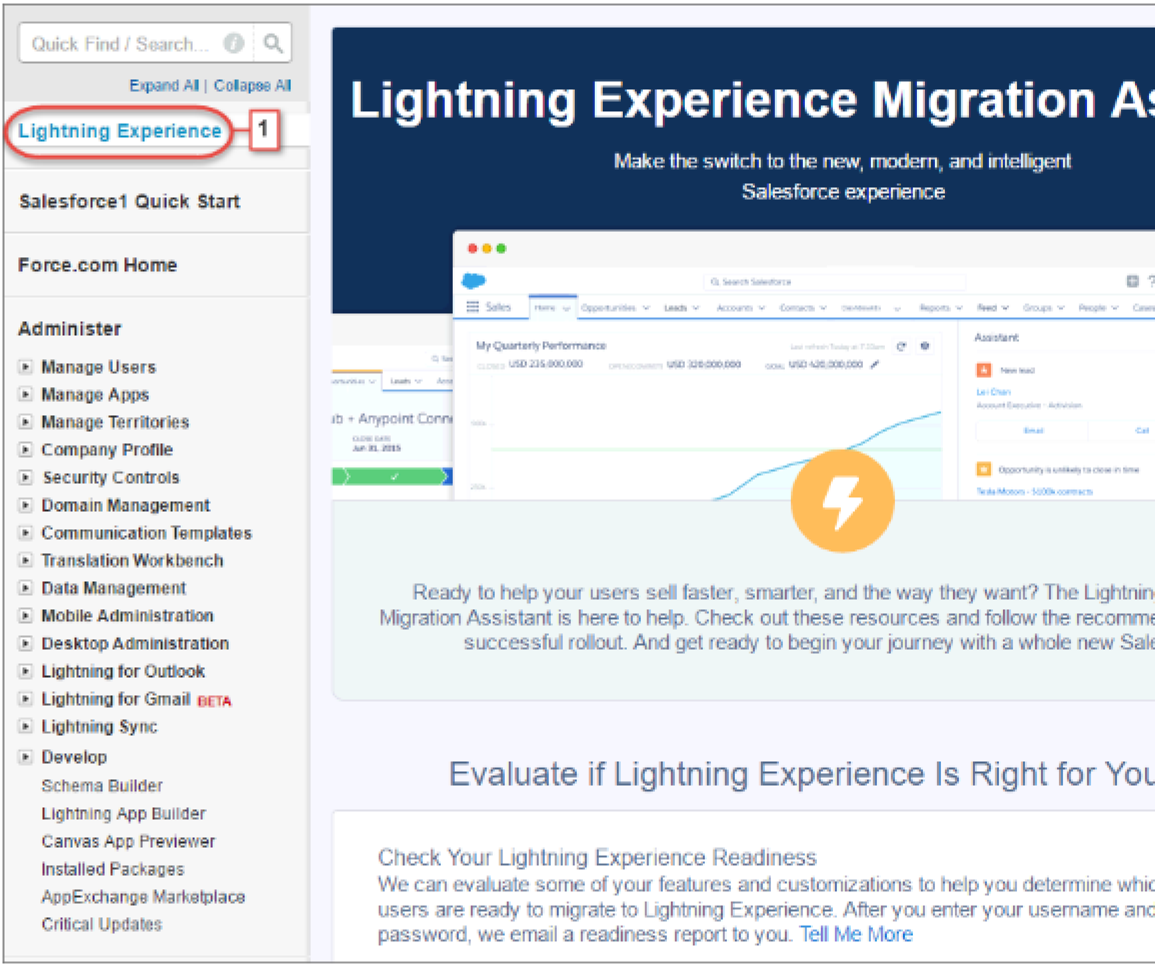
If you're brand new to Salesforce, you can skip this next step and head down to the end of the unit.

Perform a Gap Analysis

Performing a gap analysis is where you get into the nitty gritty details of your Salesforce implementation and find out what will and won't work when you move to Lightning Experience. Of all the steps for your rollout, this one is probably the most critical because it'll help you know what workarounds you might need to put in place. Then, you can educate your company on what they gain and what they might miss once you make the transition.

Luckily, we have the perfect tool for this part of your rollout. Meet the Lightning Experience Readiness Check!

Access the Readiness Check from the Lightning Experience (1) menu in Setup. Click evaluate (2) to set the tool in motion.



Lightning Experience Migration As

Make the switch to the new, modern, and intelligent Salesforce experience

Ready to help your users sell faster, smarter, and the way they want? The Lightnin Migration Assistant is here to help. Check out these resources and follow the recomme successful rollout. And get ready to begin your journey with a whole new Sale

Evaluate if Lightning Experience Is Right for You

Check Your Lightning Experience Readiness

We can evaluate some of your features and customizations to help you determine whic users are ready to migrate to Lightning Experience. After you enter your username and password, we email a readiness report to you. Tell Me More

Then, we email you your readiness assessment on the features and customizations we evaluated. We also give you our recommendations on taking next steps. The readiness check evaluates several of your Sales Cloud features, including unsupported tabs, objects, custom buttons, and links.

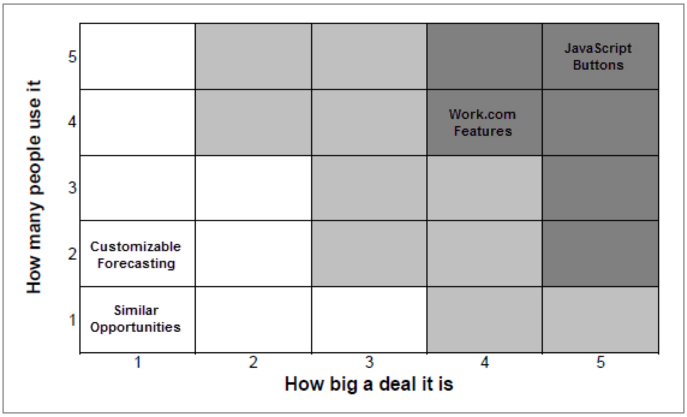
Tip: If you use Enterprise Edition or above, identify which teams are the right fit for a possible pilot of Lightning Experience.

We've also included a sample gap analysis checklist in the enablement pack for you to use, highlighting many of the key areas which you should investigate when conducting your analysis.

Present Your Findings

After you conduct your gap analysis, assess the impact. This is one of those areas where it pays to have a team! Work with your stakeholders to identify the severity of each gap. You can use several methodologies when assessing impact, including assigning a numeric value to each gap, or plotting the gaps on a simple chart.

For example, you could use a risk severity matrix like this one to categorize any gaps you've found in your analysis. (There's a sample in the enablement pack.) If you're not using similar opportunities and have limited use of customizable forecasting, those would plot low. Conversely, if you have several JavaScript buttons and your org makes heavy use of Work.com features, those would plot high.



After you and your stakeholders have assessed the impact of each gap, you are ready to present your findings to the entire steering committee and your executive sponsor. This meeting is where you showcase all the benefits of Lightning Experience one more time, and highlight the gaps you've found. Come prepared to advise when you think your company should move to Lightning Experience, the amount of time and resources required for the move, and your proposed launch date.

When the Show Is a Go

When you get the green light to proceed, get started on the next steps right away. Executing on a project is as much about preparing your users for the new experience as it is about the technical steps to implement. In the next unit, we'll help you craft a rollout strategy and then execute it in style. It's time to go live with greatness!

CHAPTER 22 All Systems Go

In this chapter ...

- Where to Start
- Plan Your Rollout
- Decide Who Gets Lightning Experience
- Unleash the Power of Super Users
- Create a Chatter Group for the Rollout Team
- Pick a Launch Date
- Create a Project Schedule
- Define Measures for Success
- Create a Marketing and Communication Strategy
- Create a Training Plan
- Test Your Customizations and Iterate
- Ready to Start

Planning and executing your Lightning Experience project should involve process, discipline, and yes, a sense of fun! What's the point of introducing game-changing new features if your users aren't excited about the rollout?

In this unit, we'll talk about how to schedule and plan your rollout effectively, but we'll also share ideas for turning your launch into an awesome event at your company.

Where to Start

Remember our Broadway musical analogy? You're the Director, and your show just got greenlighted. So now what? Now it's time to get busy.

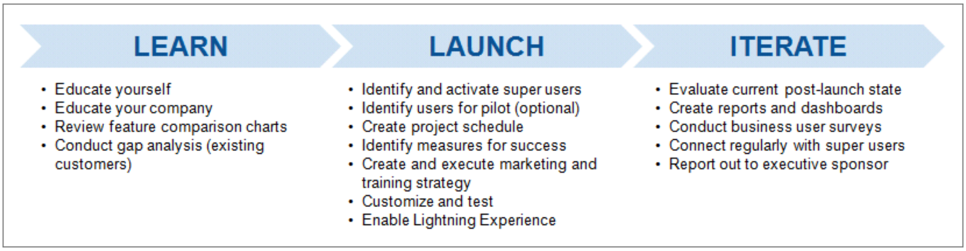
Start by listing everything you need to do before you go live, and then estimate the time and resources needed for those activities. Then select a launch date, get your team, and go do all of it. Doesn't that sound easy?

Doing all of that takes time and coordination. There are a number of methodologies for project management, too many for us to cover here. Each one has different strategies for delivering projects. Pick whichever method you like. Ultimately, the important part isn't the way you do it; it's what you do. That's what we'll talk about here.

Plan Your Rollout

There are several tasks to consider when rolling out Lightning Experience. Depending on your company, you don't have to do all of them, but consider them regardless. Check the enablement pack for a sample Lightning Experience rollout checklist.

Your rollout will likely be organized into these main buckets. You already learned about the "Learn" steps in the last unit. In this chapter, we'll look at the "Launch" category.



LEARN

- Educate yourself
- Educate your company
- Review feature comparison charts
- Conduct gap analysis (existing customers)

LAUNCH

- Identify and activate super users
- Identify users for pilot (optional)
- Create project schedule
- Identify measures for success
- Create and execute marketing and training strategy
- Customize and test
- Enable Lightning Experience

ITERATE

- Evaluate current post-launch state
- Create reports and dashboards
- Conduct business user surveys
- Connect regularly with super users
- Report out to executive sponsor

Let's dive in.

Decide Who Gets Lightning Experience

If you have Professional Edition or above, you have the option to use custom profiles or permission sets to enable Lightning Experience for a set of your users. This allows you to pilot Lightning Experience with one or more teams in your company before you go all in.

If you're thinking about going this route, consider the following:

Consider using permission sets	Permission sets are a flexible way to roll out Lightning Experience to a particular sales team or group of users. Rather than updating your custom profiles, you can create a permission set for the Lightning Experience User permission, then assign the permission set to the desired users.
Keep teams together	Put people who work closely together in the pilot together. You want people who work together seeing the same screens. If people collaborate often, they should have the same user experience, including team leaders.
Be mindful of gaps	If you have teams who are heavily impacted by gaps that you identified in your gap analysis, avoid including those teams in your pilot group. Even if the number of gaps is small, if the impact is large, consider carefully if they should be in the pilot.

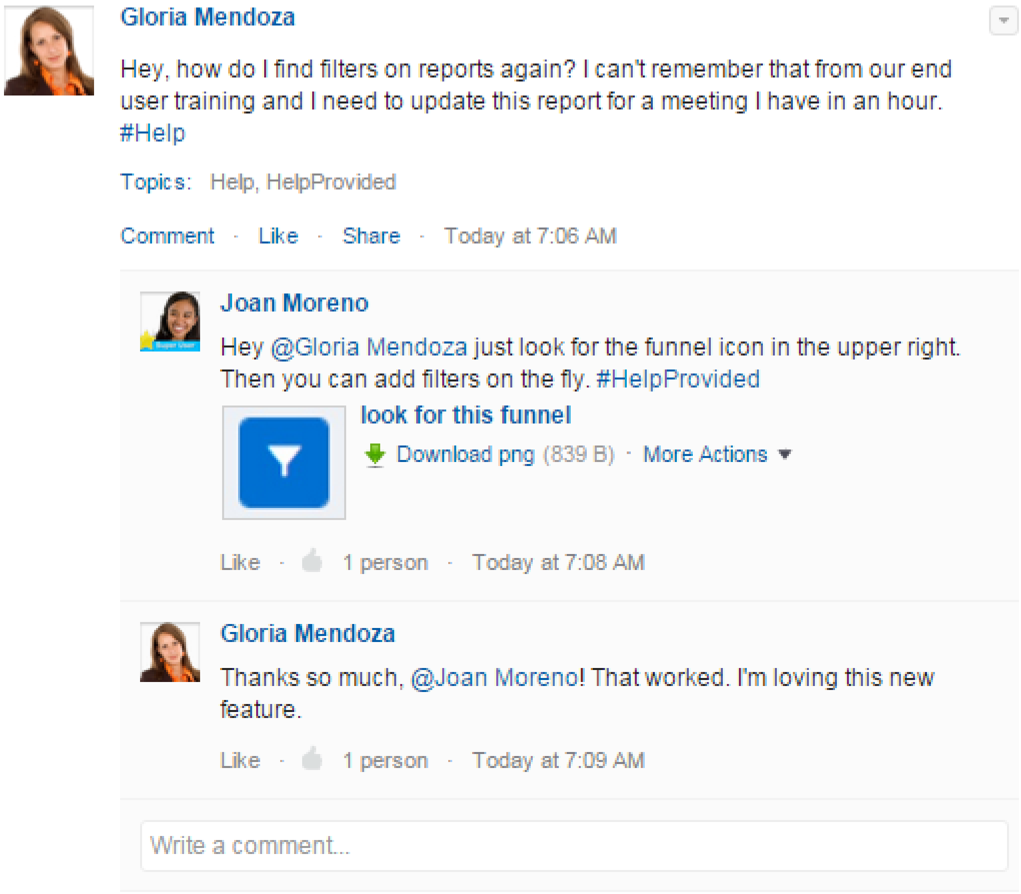
If you're using Group Edition, Lightning Experience is "all or nothing." Once you turn it on, all your users get access to the new interface. Which is obviously going to factor into your decision about when to enable Lightning Experience.

Regardless of your edition, you may be fretting about grumpy users. You know, users who have a lot going on right now and no time to learn a new user interface. Fear not! Before enabling Lightning Experience, you can specify who will switch immediately to the new interface and who will remain in the classic interface until they're ready to switch themselves.

Learn more about enabling Lightning Experience for a specific set of users in Set Up Users for Lightning Experience on page 35.

Unleash the Power of Super Users

Super users are employees who understand the vision and value of your implementation, want to help you optimize and improve what's in place, and are passionate about helping others adopt Salesforce. Often closely engaged with your employee community, super users know how well systems and processes are being adopted, and which pain points are preventing adoption. Super users are also the first people your employees go to for help, and they can be incredible in helping you answer questions and provide support.



Gloria Mendoza

Hey, how do I find filters on reports again? I can't remember that from our end user training and I need to update this report for a meeting I have in an hour. #Help

Topics: Help, HelpProvided

Comment · Like · Share · Today at 7:06 AM

Joan Moreno

Hey @Gloria Mendoza just look for the funnel icon in the upper right. Then you can add filters on the fly. #HelpProvided

look for this funnel

Download png (839 B) · More Actions

Like · 1 person · Today at 7:08 AM

Gloria Mendoza

Thanks so much, @Joan Moreno! That worked. I'm loving this new feature.

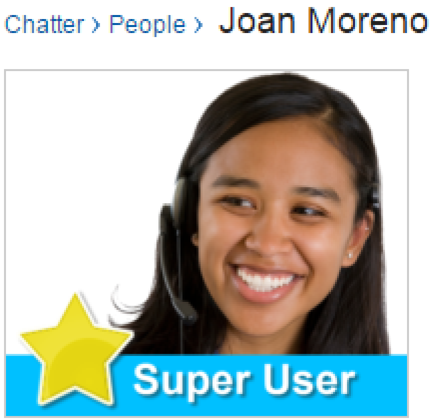
Like · 1 person · Today at 7:09 AM

Write a comment...

Super users are often natural leaders, well-respected by their peers, and can be your evangelists in the field. And when it comes to rolling out Lightning Experience, they can help make your project a success!

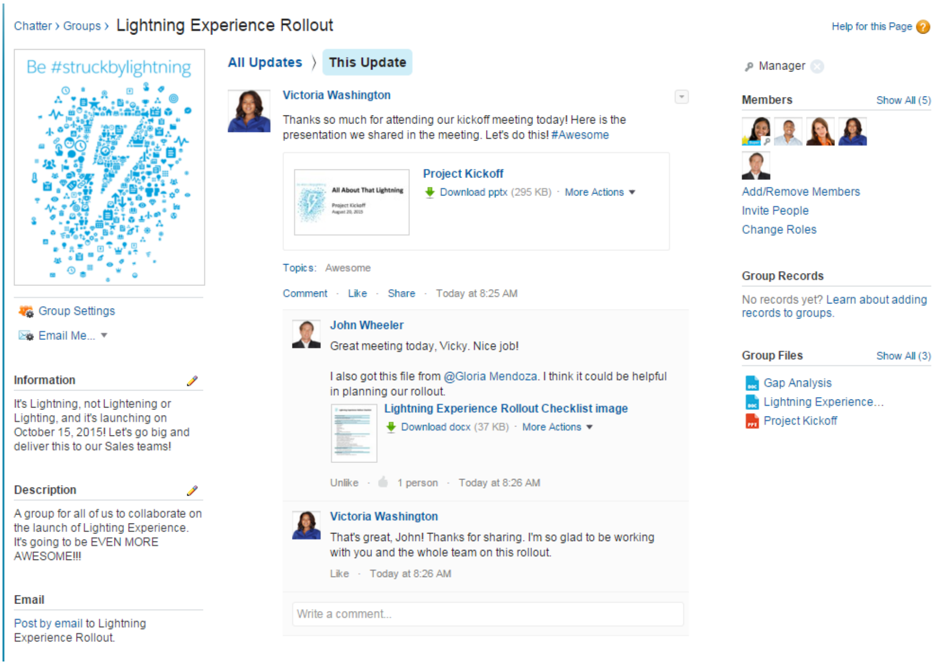
Work with your Executive Sponsor and stakeholders to identify a group of super users. Involve your super users in the rollout by giving them sneak peeks at Lightning Experience and early opportunities to train on the new technology. Seek their feedback on your marketing and communication plans, and consider asking them to train users, with a train-the-trainer approach.

Officially recognize their super user status through a special designation, like a custom icon on their Chatter profile photo, a button, a t-shirt—or all three! Make them moderators or managers of public Chatter groups for Lightning Experience. All of this helps validate their role as leaders in your employee community and empowers them to help others.



Create a Chatter Group for the Rollout Team

After you identify your super users, stakeholders, and executive sponsor, you need a place where you can all work together on the rollout. Create a Chatter group and invite in all of your team members involved with the rollout. Using Chatter, you can share files, collaborate in context, and share relevant updates with the whole team.



Information

It's Lightning, not Lightening or Lighting, and it's launching on October 15, 2015! Let's go big and deliver this to our Sales teams!

Description

A group for all of us to collaborate on the launch of Lighting Experience. It's going to be EVEN MORE AWESOME!!!

Email

Post by email to Lightning Experience Rollout.

Victoria Washington

Thanks so much for attending our kickoff meeting today! Here is the presentation we shared in the meeting. Let's do this! #Awesome

Project Kickoff

Topics: Awesome

Today at 8:25 AM

John Wheeler

Great meeting today, Vicky. Nice job!

I also got this file from @Gloria Mendoza. I think it could be helpful in planning our rollout.

Lightning Experience Rollout Checklist image

Today at 8:26 AM

Victoria Washington

That's great, John! Thanks for sharing. I'm so glad to be working with you and the whole team on this rollout.

Today at 8:26 AM

No records yet? Learn about adding records to groups.

One of the key files to share in your Chatter group is a project schedule. The project schedule is a living document that you'll want to keep updated and make accessible to your whole team, so it's ideal to store it in the cloud in your team's Chatter group.

Pick a Launch Date

Pick your launch date wisely! Think about aligning your launch to coincide with your company's sales kickoff meeting or another large company event where you can get organic exposure for your rollout. Avoid holidays and confirm stakeholder availability in advance. If a key stakeholder is on vacation for three weeks leading up to your launch date, consider revisiting your selected date!

Could Be Good Timing	Not So Much
Sales kickoff	Holidays
Company meeting	End of quarter/fiscal year
Low season (if applicable)	When key stakeholders are on vacation

Take this opportunity to review your existing Salesforce roadmap and clear any projects that might compete with your rollout. Work with your executive sponsor to clear any roadblocks, such as competing non-Salesforce projects that require resources you need for your rollout. If your rollout looks especially complex, you may need to advise your company to put other projects on hold until this project is complete.

Create a Project Schedule

There are apps on the AppExchange or software programs designed to help you manage all the milestones and tasks associated with your project. You may already have a favorite program or app that you like to use. Ultimately, you need a place to track these details:

- Task name
- Task owner
- Task dependencies (does the task depend on any other task or resource?)
- Task duration
- Task start date
- Task end date
- Task status

You might also want a place to add notes or comments.

Rolling out Lightning Experience could take you anywhere from a couple of weeks to a couple of months. The time it takes can depend on several factors, including the complexity of your organization, the size of your user population, your company's approach to change management, or if you're a new customer implementing Salesforce.

In any project, you need to be flexible either on the go-live date, the scope, or the resources allocated. Here's the first part of a sample project schedule with a fixed go-live date and scope, for a company with a straightforward migration path and a small group of users.

Check out the enablement pack for the full sample schedule. In this example, the time to go live is one month from the start date, with a launch date of October 15. There are two weeks of post-launch activities following the launch before the project officially closes, at which point you move into maintaining and iterating on the solution provided.

#	Task name	Owner	Status	Dependencies	Duration	Start	End
1.0	**Educate yourself**						

#	Task name	Owner	Status	Dependencies	Duration	Start	End
1.1	Complete Trailhead modules	John	Complete	N/A	1 day	9/5	9/6
1.2	Sign up for Developer Edition organization	John	Complete	N/A	1 day	9/5	9/6
1.3	Read the *Lightning Experience Guide*	John	Complete	N/A	2 days	9/6	9/8
1.4	Review comparison charts	John	Complete	N/A	1 days	9/6	9/7

One of the key items in phase three of the schedule is to identify measures for success. This is how you'll ultimately know if your project was successful, based on the criteria you define for what success looks like. Let's talk about that next.

Define Measures for Success

Work with your executive sponsor and stakeholders to determine how you want to measure success. Document current pain points and look for ways to measure improvements in these areas. For example, you could look for productivity gains, data quality gains, or financial goals, such as:

- 20% reduction in opportunities with no follow-up tasks
- 15% increase in calls logged
- 5% increase in lead conversion rate

You could also measure success based on employee or customer sentiment, using a survey app from the AppExchange to collect feedback, or simple Chatter polls to survey employees quickly.

In each instance, conduct a baseline survey or take an analytic snapshot in order to measure any increase or decrease following your go-live.

We'll discuss methods for measuring success in the next unit. At this stage, work with your project team to outline the specific measures you want to monitor.

Create a Marketing and Communication Strategy

This is the part of the rollout where you can inject some fun into the project, and where you can go live with greatness. Consider making your go-live into a true event and use simple marketing strategies to build buzz and excitement about the coming launch for weeks in advance.

No matter your budget, you can use your creativity to market your go-live. Don't forget to ask for help from internal teams, including customer care, training, and support teams. Launch ideas include:

- Send a weekly email drip campaign highlighting the coming launch and a "feature of the week" (enablement pack).
- Create a topic in Chatter for all your communication updates to drive momentum and buzz.
- Have a raffle with prizes, such as gift cards, a free day off, or lunch with an executive.
- Host a launch party with cupcakes or cake.
- Order swag and branded items to distribute on the day of your go-live.

Include key communication milestones in your marketing strategy. For example, make sure that the entire company gets an official communication on the day of the launch, as a Chatter post or email from your executive sponsor, VP of Sales, or CEO.

Here's a sample communication plan:

4 weeks prior	Email from executive sponsor
4 weeks prior	Chatter group created
3 weeks prior	Email drip campaign #1
3 weeks prior	Super users identified and announced
2 weeks prior	Email drip campaign #2
2 weeks prior	Raffle announced
1 week prior	Email drip campaign #3
1 week prior	Email from company President
1 day prior	Email with reminders, instructions, and where to get help

Day of Go-Live	Chatter post from company CEO & launch party

Create a Training Plan

As part of your rollout, prepare your users for the changes in their user experience.

First, direct your users to Trailhead! To help get your users up to speed, we've put together two learning experiences for them to use.

- Salesforce User Tour
- Sell Lightning Fast with Salesforce

Depending on how customized your user interface is or how complex your processes are, you may also want to conduct end user training.

When you're developing your training plan, consider these questions:

Training goals	What is the specific outcome you want to achieve with this training?
Trainer	Who will conduct the training?
Trainees	Who needs to be trained?
Training methods	What will you use to conduct the training? What materials need to be developed?
Training location	Will you train remotely or in-person?
Training metrics	How will you determine if the training was successful?

Consider using a train-the-trainer model with your super users to help you extend your reach. Rather than training all of your users personally, you can train your super users and have them conduct user training. This train-the-trainer model also helps reinforce to users that they should go to the super users with any questions post-launch.

Also consider follow-up training sessions or office hours after your launch to help reinforce the right behaviors and keep your users current. If you have a support team, involve them in the training too, so they can be prepared for user questions.

Test Your Customizations and Iterate

For existing customers, if you already have customizations in place, we recommend enabling Lightning Experience in a sandbox and testing their behavior. For unsupported features, like JavaScript buttons, analyze what the underlying function is of each. Here's a set of questions you can use in your analysis:

- What does the customized feature do?
- What objects are affected or accessed?
- What are the resulting actions of using the customized feature?
- What is the user experience?
- Where can your user access the customized feature?

After you have the answers to these questions, you can start to map the customized feature to a possible replacement. For example:

If the customized feature does this:	**Consider using this instead:**
Creates a related record	Process Builder
Updates an existing record	Actions
Creates related records and updates existing records, with complex logic	Process Builder and Visual Workflow
Launches a screen for user input	Visual Workflow
Sends an email or creates a task	Process Builder
Launches a time-triggered process	Workflow Rules

As you work through updating these processes, work closely with your super users and users to test the replacement solutions you build. Create test plans and conduct User Acceptance Testing (UAT) to ensure features work as expected. Get a sample test plan document in the enablement pack.

Ready to Start

After you've created these plans, work with your project team, stakeholders, executive sponsor, and super users to execute them. And once you've checked all the items off your list, it's time to go live! In the next unit, you'll enable Lightning Experience and find out how you can boost your success post-launch.

CHAPTER 23 Go Live

In this chapter ...

- Engage with Super Users
- Measure Results
- Survey Your Users
- Use Reports and Dashboards in Salesforce
- Deliver an Executive Summary
- The Show Must Go On

There comes a time in every successful project where you've planned everything, executed on all those plans, and then the launch date arrives. This means it's time to go big and go live with greatness. If this is your Broadway musical, then it's opening night, and Salesforce admin, this is your time to shine!

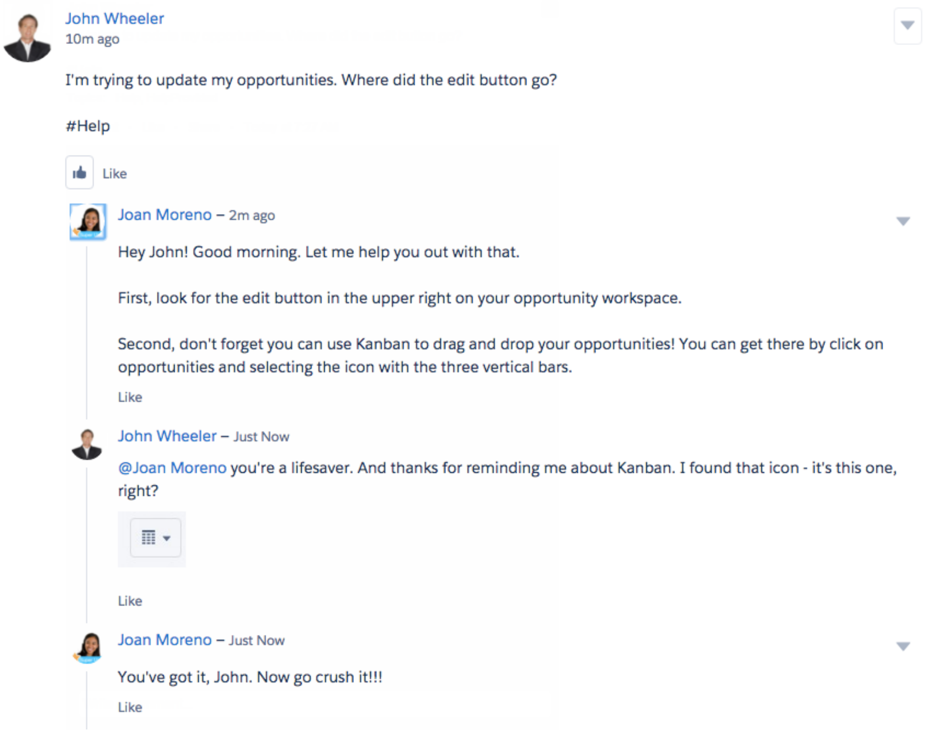
Depending on the project, the go-live step can be simple, complex, or somewhere in between. We make it easier with the Lightning Experience Migration Assistant, where you can access all the steps that need to be completed to move to Lightning Experience. Access this page from Setup in Salesforce Classic, by clicking **Lightning Experience**.

The Migration Assistant is your guide to enabling Lightning Experience. Keep in mind that there are other features we recommend that you enable before turning on Lightning Experience. Key features are covered by the Migration Assistant, and you can also review Recommended Features for Lightning Experience on page 34 for full details.

Engage with Super Users

Your super users are key to the success of your rollout. Leading up to your go-live, engage with your super users often. Depending on your training approach, you might be partnering with them as they lead user training in the field.

On the day of your go-live, super users play an essential role. Even with the most successful projects, your users are going to have questions. Those questions can come in the form of in-person conversations or questions asked on Chatter. Your super users can help you swarm on those questions, and escalate any bugs or critical issues that are discovered.



John Wheeler
10m ago

I'm trying to update my opportunities. Where did the edit button go?

#Help

Like

Joan Moreno – 2m ago

Hey John! Good morning. Let me help you out with that.

First, look for the edit button in the upper right on your opportunity workspace.

Second, don't forget you can use Kanban to drag and drop your opportunities! You can get there by click on opportunities and selecting the icon with the three vertical bars.

Like

John Wheeler – Just Now

@Joan Moreno you're a lifesaver. And thanks for reminding me about Kanban. I found that icon - it's this one, right?

Like

Joan Moreno – Just Now

You've got it, John. Now go crush it!!!

Like

But your engagement with super users doesn't end with your go-live. That's just the beginning. Consider setting up a monthly or quarterly super user forum meeting where you gather all your super users together and talk about the roadmap, feedback, and overall adoption. Create a private Chatter group with your super users to facilitate conversation year-round. And work with your super users to host weekly office hours to answer user questions.

If you're not yet convinced, here's a short list of some of the ways these incredible team members put the "super" in super user:

- Play the role of Salesforce evangelist in the field
- Swarm on questions from users, in-person and on Chatter
- Help you boost adoption
- Train users
- Share valuable insight from the field
- Report bugs and issues
- Weigh in on the roadmap
- Test new features before they go live
- Provide feedback in forums, meetings, and focus groups

As you'll see, super users provide tremendous value, especially when it comes to sharing feedback, which is critical for measuring the success of your project. Let's talk about that next.

Measure Results

When you were planning your rollout strategy, you worked with your executive sponsor and stakeholders to determine your success metrics. Those metrics might have included things like:

- 20% reduction in opportunities with no follow-up tasks
- 15% increase in calls logged
- 5% increase in lead conversion

Or, the metrics might have been based on employee or customer satisfaction, or even a mixture of both. For either approach, you need a baseline to work from to track changes in the numbers, so don't forget to take a snapshot or conduct a survey before you go live.

Let's discuss each approach in detail.

Survey Your Users

You can use several methods to survey your employees and customers.

Survey	Use a survey app to conduct a formal written survey, measuring overall satisfaction and any pain points.

Focus Groups	Bring together a group of customers or employees and ask a set of specific questions.
Poll	Create an informal Chatter poll to gather quick insights. Users can also provide written feedback in the poll's comment thread.

Using these methods, you can obtain metrics about overall satisfaction and any pain points. Following your go-live, after users are comfortably working with the new experience, you can survey again and report on any changes in the metrics.

Not sure what to ask? Here are some ideas. We're including questions that are open-ended and appropriate for focus groups, forums, and shadowing, and also questions more appropriate for a written survey:

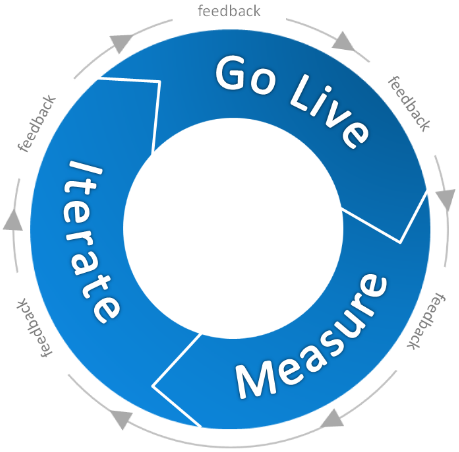
Open-ended questions	Survey questions
How do you use Salesforce?	How long have you been using Salesforce?
What do you like best about Salesforce?	Rate your overall productivity using Salesforce.
What would you change about Salesforce?	Rate your overall satisfaction with Salesforce.
What is most frustrating about Salesforce?	I have the tools I need to do my job.
What information do you need that you can't find?	It's easy to work in Salesforce.

In addition to conducting surveys, cultivate an environment where feedback can flourish. Listening to your users boosts adoption. Here are some ideas for how to encourage your users to give regular feedback.

Open Forum	Host an open question and answer session, online via Chatter or in-person, to get feedback and hear concerns.
Chatter	Use a Chatter Topic ("feedback") to collect responses from your users organically over time. Review all the posts monthly or quarterly.

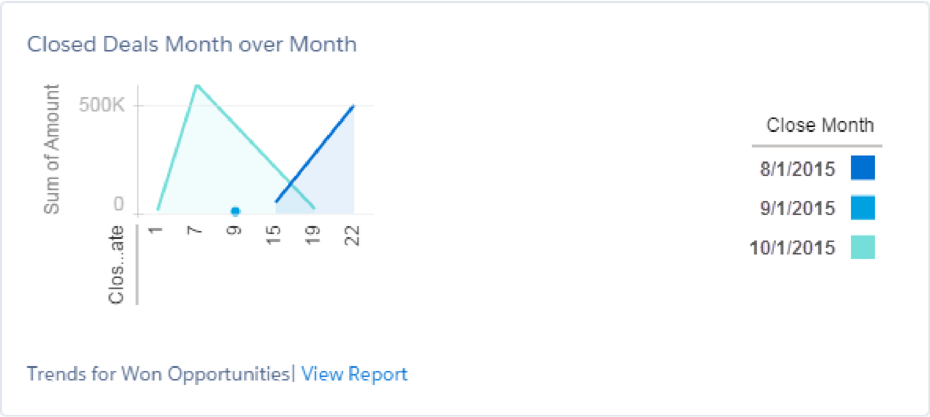
When you're receiving feedback, it's important that your users feel heard, even though you can't address all feedback. Consider starting a "You Asked, We Listened" program where you address some employee feedback items every quarter.

Whether you're doing a limited pilot or rolling out to your whole company at once, the goal is to go live, measure your results, and then iterate. Then repeat.



Use Reports and Dashboards in Salesforce

If you've decided to use Salesforce metrics for your success measures, then you can create reports and dashboards within Salesforce to track those metrics. For example, this dashboard component shows closed deals, month over month:

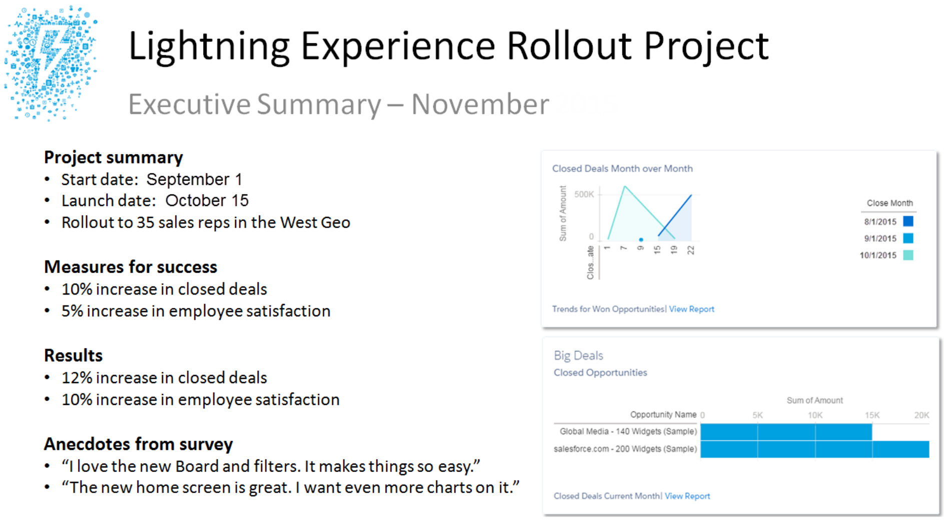


Reporting Snapshots are a way to analyze trends over time, right in Salesforce. Search for "Reporting Snapshots" in the Salesforce Help to learn more.

One of the benefits of using reports and dashboards is that they're all built within Salesforce, which makes it easy to share with your executive sponsor. Speaking of which…

Deliver an Executive Summary

Your Executive Sponsor has been your project champion from the start. Now it's time to prepare a summary for your sponsor on the overall project status and any results you're able to share. This is important for your Executive Sponsor to be able to showcase Return on Investment (ROI) for the resources that were allocated to complete the project.



Lightning Experience Rollout Project

Executive Summary – November 2015

Project summary
- Start date: September 1
- Launch date: October 15
- Rollout to 35 sales reps in the West Geo

Measures for success
- 10% increase in closed deals
- 5% increase in employee satisfaction

Results
- 12% increase in closed deals
- 10% increase in employee satisfaction

Anecdotes from survey
- "I love the new Board and filters. It makes things so easy."
- "The new home screen is great. I want even more charts on it."

Closed Deals Month over Month

Trends for Won Opportunities| View Report

Big Deals

Closed Opportunities

Closed Deals Current Month| View Report

When you're preparing an executive summary, follow these best practices:

- Keep it to 1-2 pages maximum.
- Showcase metrics and results.
- Highlight any noteworthy anecdotes.
- Share lessons learned.
- Note any next steps still planned.

Looking for a sample Executive Summary document? Sorry, we didn't make one. Just kidding! Check the enablement pack.

The Show Must Go On

Congratulations! You now have the tools you need to roll out Lightning Experience. Hopefully, you've also picked up some tips along the way to help you with other rollouts beyond Lightning Experience.

Whether you're piloting with a group of users or rolling out to your entire company at once, this is your chance to go big. Good luck with your rollout!

LIGHTNING EXPERIENCE DEVELOPMENT

CHAPTER 24 User Interface Development Considerations

In this chapter ...

- Raising the Bar for Web App User Interfaces
- Classic Visualforce
- Visualforce as a JavaScript Application Container
- Lightning Components
- Choosing the Right Tool for the Job
- Choosing the Right Tool for Your Organization
- Migrating to Lightning Components

Lightning Experience is a brand new, amazing opportunity for developers on the Force.com platform. In this module we're going to look at many of the aspects that make Lightning Experience different for developers. Before we get to that, though, we want to have a quick chat about a couple higher-level concerns.

Let's start with something we want to be completely honest with you about: Lightning Experience isn't finished yet. If you've visited some of the other Lightning Experience Trailhead modules, you already know there is plenty of work for us to do just building out the basic Salesforce application. On the Force.com side of things, we have plenty to do as well. Most things are working great, but there are still some things we haven't gotten to yet.

About this we want you to know two things: First, we're very hard at work turning Lightning Experience into a complete development platform for your apps. And—Safe Harbor, yadda yadda yadda—you'll be getting important missing features in the next release, and the next, and so on.

Second, we very much want your feedback. Tell us what's working great to make us feel good, and tell us about what's not, to keep us working hard, to shape priorities, and to help us make sure Lightning Experience becomes the best cloud-based development platform you can imagine.

Finally, we want to answer some really important questions that you're asking us about where you should put your efforts as you grow your existing apps and build new ones. Put simply, what's the future of user interface development and custom apps on the Force.com Platform?

We're glad you asked! We're so excited about the answer, we're going to take the rest of this unit to go beyond just answering the question, and try to give you insights into our thinking.

Raising the Bar for Web App User Interfaces

Over the last decade or more, we've all seen the bar continually be raised for user experience in web applications. Users expect responsive, fully featured, and highly interactive, immersive experiences literally at their fingertips.

We first saw this in single-purpose apps. Services like Google Maps introduced direct manipulation of user interface elements. Analytics applications brought dynamic, interactive chart drill-downs. Even the humble sign-up or log in form comes with dynamic error feedback when users enter invalid data, animations, transitions, and more. Interactivity is no longer a novelty, it's the norm.

And the scale has grown. The expectation for individual components has quickly spread to the core application experience. Today web applications feature things like sliding menus, animated page transitions, and dynamic master-details. There's also app-style elements such as overlays and modal windows. The difference between native applications and web applications has never been smaller.

So what does this mean for Salesforce?

The traditional Salesforce experience, lovingly known as Salesforce Classic, is an example of a **page-centric** web application model. It's great for basic functionality, but it's challenging to deliver the new, more dynamic experience that users expect. Fundamentally, this is because it relies on the server to generate a new page every time you interact with the application.

To deliver a more interactive experience, you need help from JavaScript on the client-side. In this new **app-centric** model, JavaScript is used to create, modify, transform, and animate the user interface rather than completely replacing it a page at a time. This model is exciting, interactive, and fluid. This is the new Lightning Experience.

Both the page-centric and app-centric models are here to stay. A quick look around the web is enough to show that most web applications take advantage of both approaches. Combining these models lets applications deliver the right type of experience for the right use case.

Let's take some time to explore the different options that the Salesforce Platform offers for these models.

Classic Visualforce

Visualforce is a robust, mature platform for building page-centric apps. The Visualforce framework provides a robust set of tags that are resolved on the server, and that work alongside standard or custom controllers to make database and other operations simple to implement.

Let's review some basics.

UI Generation

Server-side

Data and Business Logic

Apex standard or custom controller

Workflow

1. User requests a page
2. The server executes the page's underlying code and sends the resulting HTML to the browser
3. The browser displays the HTML
4. When the user interacts with the page, return to step one

Pros

- Tried and true model
- Easy to implement for greater productivity
- Naturally splits large applications into small, manageable pages
- Has built-in metadata integration

Caveats

- Limited interactivity (aside from added JavaScript functionality)
- Higher latency, which degrades mobile performance

Visualforce is conceptually similar to other page-centric technologies like PHP, ASP, JSP, and Ruby on Rails. Salesforce's rich metadata infrastructure makes Visualforce a productive solution. The standard controller makes it easy to access objects directly and via relationships without executing a single query. Other metadata-aware components are similarly plug-and-play: add markup to a page and you're done. These capabilities are alive and well on the platform, and they're still suitable for many use cases.

Visualforce as a JavaScript Application Container

When you think about it, Visualforce pages are just HTML pages with extra tags resolved by the server. As a result, you can use an empty Visualforce page as a container for a JavaScript application. In this scenario, you don't use Visualforce tags to build your user interface. Instead, you load your JavaScript application in an empty page. Then the user interface is generated on the client-side by the JavaScript application. These applications are generally referred to as *single-page applications*, or SPAs, and are often built using third-party frameworks like AngularJS or React.

Let's review some basics.

UI Generation

Client-side (mostly JavaScript)

Data and Business Logic

Remote Objects or JavaScript Remoting, Apex controller

Workflow

1. The user requests the "empty" Visualforce page containing a page skeleton and JavaScript includes
2. The page is returned to the browser
3. The browser loads the JavaScript application
4. The JavaScript application generates the UI
5. When a user interacts with the application, the JavaScript modifies the user interface as needed (return to the previous step)

Pros

- Enables highly interactive and immersive user experiences

Caveats

- Complex
- No built-in metadata integration
- Lack of an integrated developer experience. The Force.com Developer Console doesn't explicitly support these JavaScript applications. Typically, you have to load them as static resources, and that's a cumbersome experience.

We want to be clear. If you can live with the caveats we've described, this is a perfectly good way to build interactive applications today. It's the reason we originally built toolkits like Remote Objects and JavaScript remoting. If you're a confident AngularJS or React or other JavaScript framework developer, your expertise will take you a long way developing apps for Salesforce with the tools you know.

But, if you're open to new things, we think we have some great ideas for what the next level is in web-based application development.

Lightning Components

Lightning components are part of the new Salesforce user interface framework for developing dynamic web applications for desktop and mobile devices. They use JavaScript at the client-side and Apex at the server-side to provide the data and business logic.

Let's take a look at an overview.

UI Generation

Client-side (JavaScript)

Data and Business Logic

Lightning Data Services, Apex controller

Workflow

1. The user requests an application or a component
2. The application or component bundle is returned to the client
3. The browser loads the bundle
4. The JavaScript application generates the UI
5. When the user interacts with the page, the JavaScript application modifies the user interface as needed (return to previous step)

Pros

- Enables highly interactive and immersive user experiences
- Aligns with Salesforce user interface strategy
- Built on metadata from the foundation, accelerating development
- Both the Force.com IDE and the Developer Console support Lightning components, providing an integrated developer experience

Caveats

- Steeper, longer learning curve compared to Visualforce
- Higher complexity than Visualforce—you're building an application, not a page
- Since Lightning components are new, there are still some features that aren't supported
- There are a limited number of out-of-the-box components

We need to be straight with you. These caveats aren't small considerations. We'll talk about how they apply to your organization in a bit. But! We're working hard—really hard—on reducing the size of these considerations. We're super-excited to bring improvements to you, as soon as we can.

Choosing the Right Tool for the Job

Visualforce has been around for a while, it's a mature, well-understood platform for building your apps. It's not going away. Lightning Components is the new kid on the block. It's got a lot going for it, but, well, you know. It's a stranger to you right now.

Here's the thing: You don't have to choose one or the other.

Think of the page-centric and app-centric models as tools in your development tool belt—one isn't always better than the other, and you'll get the most out of them if you understand their strengths and their trade-offs. Use the right tool for the job at hand.

Here are some guidelines to help you decide—but remember, you're The Decider. In the end, use the tool that feels right *to you*. Also, keep in mind that tools evolve. These guidelines will evolve too.

Job	Recommendation
I'm Developing for Lightning Experience	We highly recommend Lightning Components. Lightning Experience was *built* with Lightning Components, and the two fit together like hand and glove. You can certainly use Visualforce if you have existing code or a project underway. Visualforce for Lightning Experience is fully supported, with a few constraints. But you won't find a better tool for Lightning Experience than working in its native language, Lightning Components.
I'm Developing for the Salesforce1 Mobile Application	We recommend Lightning Components. Visualforce characteristics, especially the page-centric orientation, can be a poor match for mobile apps with limited, high-latency network connections and limited compute resources. Lightning Components, by contrast, was designed *specifically* to handle this context. Both Visualforce and Lightning Components use similar tag-based markup. For example, Visualforce markup for an input field is `<apex:inputText>` while for LightningComponents it's `<ui:inputText>`. So what's the difference? Well, Visualforce processes markup tags on the Salesforce server. Lightning Components process markup on the client. The advantage of processing on the client is that the HTML block for the entire page isn't passed back and forth between the client and the server with every interaction. With a few exceptions, Lightning Components are better for Salesforce1 development. Some cases call for Visualforce as a JavaScript application. See the Lightning Components Developer's Guide for more information.

Job	Recommendation
I'm Building a Page-Centric Experience with Limited Client-Side Logic	Use Visualforce pages to ensure development velocity and manageability.
I'm Building an Interactive Experience with JavaScript to Meet User Experience Requirements	Use Lightning Components but refer to limitations documentation first.
I'm Committed to a JavaScript Framework Such as AngularJS or React	Use a Visualforce page as a container for your third-party framework, such as AngularJS or React, and your application.
I'm Enabling Non-Developers to Build Apps by Assembling Standard or Custom Components	Use Lightning Components to create custom components that can be used in the Lightning App Builder.
I'm Building an Interactive Experience with JavaScript and I Need a Third-Party Framework	Use a Visualforce page as a container for your third-party framework, such as AngularJS or React, and your application.
I'm Adding User Interface Elements	For example, say you want to add a tab to a record home. This task is a simple drag-and-drop in Lightning App Builder. Use Lightning Components to create custom user interface elements.
I'm Building a Community for Customers	Use Community Builder to create a Lightning-based community leveraging Lightning Components.
I'm Building a Community for Partners	Use Community Builder to create a Lightning-based community leveraging Lightning Components.
I'm Exposing a Public-Facing Unauthenticated Website	Use Community Builder to create a Lightning-based site leveraging Lightning Components.
I'm Rendering Pages as PDFs in My Application	Use Visualforce. Lightning Components don't support rendering as PDF output yet.
I'm Adding to an Existing Project with Lots of Visualforce Pages	Continue to use Visualforce. Consider opportunistically moving elements to Lightning Components using Lightning Components for Visualforce.

Job	Recommendation
I'm Committed to Investing in the Latest Technology	You're right there with us! Dive in to Lightning Components. Start with the Lightning Components Basics Trailhead module.
I'm Starting a Brand New Project	Use Lightning Components. If you're not familiar with them, there's no better time than now to learn!

Choosing the Right Tool for Your Organization

When you think about choosing a tool, it's important to focus on more than just the job at hand. You also want to consider your organization as a whole and your role within your organization. Let's look at how some different roles can best leverage Salesforce's development models.

Role	Recommendation
ISV Partner	Start using Lightning components for new apps or new features in existing apps. Package these units for subscriber use in both Salesforce Classic and Lightning Experience.
SI	Start using Lightning components for new implementations. For in-progress implementations, continue using Visualforce.
Professional developers who are JavaScript gurus and experienced with Visualforce	Continue using Visualforce with your preferred JavaScript framework. Explore Lightning components and consider switching down the line.
Citizen developers who use standard Visualforce components for pages	Continue using Visualforce, but consider checking out Lightning App Builder.
Point-and-click admins	Use Lightning App Builder to create apps and customizations. Buddy up with developers and partners to build custom Lightning components.

Migrating to Lightning Components

Here's the good news. Despite the shift to Lightning Experience and an increased focus on Lightning components, your Visualforce pages still work on Salesforce. It doesn't matter if you're using the new interface or your old friend Salesforce Classic—Visualforce is able to work with both. You don't have to convert any existing Visualforce pages to keep using them for a long time.

But, because web applications are taking more advantage of the app-centric model, we encourage all Salesforce developers to learn at least the basics of Lightning components. You'll want to use these components in your future development work.

Now is the perfect time for you to take the first steps. Features such as the Lightning App Builder and Lightning Components for Visualforce let you "dip your toe in the water," and try using as little as a single Lightning component in a new or existing page. You can use these embedded components in both Lightning Experience and Salesforce Classic. It's so easy, you'd be nuts not to give it a try.

We know that migrating to a new development framework is daunting. But we're here for you. This trail is loaded with all the tips, tricks, and gotchas that you need to successfully adopt Lightning Experience development.

CHAPTER 25 Visualforce and Lightning Experience

In this chapter ...

- What Works
- What Works, But Needs Updating and Testing
- What Doesn't Work
- That Look-and-Feel Thing

Lightning Experience is a whole new world, and we hope you think it's exciting. Behind the Lightning Experience user interface is a new way of delivering the Salesforce application, one that brings significant changes to the way Visualforce apps run. For the most part, your Visualforce apps should "just work," but there *are* some things you should know before you make the jump to Lightning Experience.

The technical details of how Lightning Experience is built and how it runs Visualforce apps are really cool, and important for actual development work. When you're ready for those details, the Visualforce & Lightning Experience module will show you the way. But here we're going to stay at a high level, and divide things up into what works and what doesn't, what you'll want to update, and other issues that will help you plan your Lightning Experience migration development effort.

What Works

The list of what parts of Visualforce work in Lightning Experience is pretty long. It's very nearly as long as the entire Visualforce features list. So before we get to the things that *aren't* on the list, let's think positive, and check off some of the many things you can count on.

First of all, the fundamental mechanisms of how Visualforce works remain the same. Whether your pages use the standard controller, custom controllers, JavaScript remoting, or Remote Objects, your connection to Salesforce works the same.

Note: If you're using JavaScript extensively, or if you're using other APIs to access Salesforce, you might have some work to do. We'll get to that.

Second, Visualforce markup remains the same. There are a few tags and attributes that behave differently in Lightning Experience, and a very few that we recommend against using or that just don't work. But otherwise the way you write code for Visualforce pages and components is unchanged.

Third, most of the ways you can use Visualforce to customize your organization work just fine in Lightning Experience—though as you can no doubt imagine, with an all new user interface, these customizations have moved to different places.

Let's dive just a bit into the specifics of these customizations. All of the following work just fine in Lightning Experience, simply moving to a new location in the user interface.

- Creating custom tabs and apps with Visualforce pages.
- Creating custom buttons and links that lead to Visualforce pages.
- Creating custom actions that open with a Visualforce page.
- Overriding standard actions with Visualforce pages (with one exception that we'll get to later).
- Creating flows that use Visualforce pages.
- Packaging Visualforce pages and components.

The changes in the user interface vary from minor to significant. Features customized with Visualforce move automatically when users change between Lightning Experience and Salesforce Classic. You might need to give your users an initial orientation, but after that, they'll be happy as clams.

There are other features, such as Visualforce email templates, that use Visualforce code behind the scenes. These aren't surfaced in the user interface directly, and so they remain unchanged.

For a screenshot-heavy visual tour of where various features have moved to, see the Using Visualforce in Lightning Experience unit in the Visualforce & Lightning Experience module.

What Works, But Needs Updating and Testing

The environment in which a Visualforce page runs when Lightning Experience is enabled is different than the standard Visualforce request. The technical details get pretty complex, but the simple version is that in Lightning Experience, Visualforce pages are embedded in an HTML iframe that's displayed inside the Lightning Experience app.

This change has a number of consequences that mostly have to do with JavaScript and accessing external apps. You'll want to review your code and verify a few things before you certify your Visualforce pages for use in Lightning Experience. We're saving the nitty gritty details for the Visualforce & Lightning Experience module. For now, let's just check in at a high level so you can start to scope your review.

For starters, if you have pages or apps that use JavaScript, you'll want to review the behavior of the code. In particular, your code can't directly access the `window` global object. You can still get at it with a minor code change if you *really* need to, but there are probably better ways to accomplish those tasks using Lightning Experience app APIs instead. In particular, code which sets `window.location` directly should definitely be revised to integrate with the Lightning Experience navigation stack.

Similarly, code that assumes that it has access to the entire environment is in for a rude surprise. It still has access to the *Visualforce* part of the document, but *not* the full Lightning Experience app. That will be fine for many apps, but for those that want to be totally in charge, there's going to be some work for you to do.

If your pages make use of iframes themselves, either with `<apex:iframe>` or static HTML, being embedded into another iframe could cause some issues. In many cases "turtles all the way down" is fine. Just make sure you do extra testing here.

If your pages embed a Force.com Canvas app, and especially if you've used the Canvas APIs to integrate the app into Salesforce, allocate time for thorough testing as well. Canvas apps use an iframe, and while correctly behaving code should just work, we all know how common perfect code is in the real world.

Pages that use Remote Objects and JavaScript remoting work without requiring updates to authentication code. However, if your pages use other Salesforce APIs you might need to adapt your authentication code to make the right cross-domain request, or otherwise adjust to the new environment.

All of the above sounds both vague and hard to do but, in truth, the amount of code you're likely to need to change is small. And again, the details for developers are available in the Visualforce & Lightning Experience module.

What Doesn't Work

And so we come to the, shall we say, less pleasant part of our conversation. Fortunately, the list of what doesn't work in Visualforce for Lightning Experience is short, and we can get through it quickly.

Perhaps the most significant change, in terms of things that might be hard to work around, Visualforce overrides of standard actions are slightly different in Lightning Experience compared to Salesforce Classic. Any override for the object list action won't be accessible in Lightning Experience.

Specifically, there are six standard actions you can override for an object in Salesforce Classic:

- Object tab
- Object list
- Record view
- Record edit
- Record create
- Record delete

In Lightning Experience the first two actions have been combined into one page, object home. Object home is similar to object list, with some elements of object tab, like recent items, added. Others, such as reports or tools, have moved to other parts of the user interface.

Regardless of the user interface settings in your organization, both object tab and object list are available to be overridden in Setup. Overriding the object tab action overrides the object home page in Lightning Experience, as expected.

However, when in Lightning Experience, the object list action isn't accessible in the user interface, so there's no way to fire it. If your organization has overridden the object list action for any object, that functionality won't be available when users are using Lightning Experience. If there are essential features in that override, you'll need to find another way to make them available.

On a bit smaller scale, the `showHeader` and `showSidebar` attributes of `<apex:page>` have no effect on Visualforce pages when displayed in Lightning Experience. The standard Salesforce Classic header and sidebar are always suppressed, and there isn't a way to suppress the Lightning Experience header and sidebar.

A number of related lists available in Salesforce Classic aren't supported in Lightning Experience. The `<apex:relatedList>` component isn't a way around this limitation. Good try, though!

And, coming down to the really minor issues, rendering Visualforce pages as PDFs works exactly as in Salesforce Classic, without any of the Lightning Experience visual design. This is probably what you want anyway, but if you wanted to render pages into PDFs that include the Lightning Experience design, that's not possible today.

That Look-and-Feel Thing

The first thing you notice about Lightning Experience is the gorgeous, all new visual design. And if you've been developing Visualforce pages for a while, your next thought might be, how will my Visualforce pages look in Lightning Experience. The short answer is…well, let's sit down for this part, OK?

The short answer is, with the exception of suppressing the Salesforce Classic header and sidebar, and of being framed by the Lightning Experience user interface, Visualforce pages display unchanged in Lightning Experience.

Specifically, the HTML that's rendered by the built-in Visualforce components doesn't change when the page displays in Lightning Experience, and the Salesforce Classic stylesheets those components use is, by default, loaded by the page. The effect is that pages that use `<apex:inputField>`, `<apex:outputField>`, the `<apex:pageBlock>` components, and other coarse- and fine-grained components that match the Salesforce Classic visual design, still match that visual design. You get a little slice of Salesforce Classic in the middle of your Lightning Experience.

If, however, you've used Visualforce components that are relatively unstyled, or your own components and markup, and developed your own stylesheets instead of using the default Salesforce styles, your pages also appear unchanged, retaining the styling you worked so hard to develop.

In other words, in Visualforce for Lightning Experience we've favored stability in the visual design of existing pages, instead of trying to adapt them dynamically to Lightning Experience.

That said, if you're as excited about the new visual design as we are, there are ways for you to adopt that styling to a lesser or greater degree, depending on how much work you want to put in. We won't cover it here, but there's a whole unit, Understanding Important Visual Design Considerations, that shows the range of possibilities and the techniques to use to achieve them.

CHAPTER 26 Lightning Components in Lightning Experience

In this chapter ...

- Considerations for Use

By now you've read the word "Lightning" so many times it's probably lost all meaning. Worse, we've been talking so much about both "Lightning Experience" and "Lightning components" that maybe the two terms are blurring together. Let's clear up the relationship between the two.

Remember all that information about developing following either a page-centric or app-centric model? Salesforce Classic uses a page-centric model, but Lightning Experience uses an app-centric model. It's made up of—you guessed it—components.

You can probably see where this is going. Lightning components were designed with Lightning Experience in mind. As the core Salesforce app shifts to the app-centric framework, we want you to shift along with us. We want you to think about developing on the platform in a whole new way.

You might have developed some Lightning components in Salesforce Classic. You can still use the old interface with Lightning components and all your existing component functionality transfers seamlessly into Lightning Experience.

If you haven't worked with Lightning components yet, don't worry. There's a Trailhead module, a quickstart guide, and a full developer's guide so that you can start developing in no time. Before you dive in to the technical details, let's take a second to review some basic advantages of Lightning components:

Out-of-the-Box Component Set

Salesforce provides a number of components to bootstrap your app development.

Performance

The component framework leverages a stateful client (using JavaScript) and a stateless server (using Apex). This structure allows the client to call the server only when absolutely necessary. With fewer calls to the server, your apps are more responsive and efficient.

Event-Driven Architecture

Events are key to the Lightning component framework. Components listen to application and component events and respond accordingly.

Rapid Development

The simple markup and pre-made components mean that you can get applications out the door faster than ever. Particularly if you're comfortable with Visualforce markup, learning component markup is a breeze.

Device-Aware and Cross-Browser Compatibility

A huge advantage of Lightning components is that you don't have to worry about compatibility across devices and browsers. The component framework handles this work for you.

Considerations for Use

We've already covered a lot of the considerations for using Lightning components. You probably don't want to switch to Lightning components with in-progress Visualforce projects. You also want to stick with Visualforce if you want to do things like render PDFs on a page. Again, Visualforce still works like it used to and continues to be a foundational part of developing on the Salesforce platform. Lightning components are still in their infancy and not all the features you're used to in Visualforce are fully supported yet. We've released several documents outlining the specific limitations of Lightning components so you can decide if they're right for your immediate development work.

We've also covered situations where you should consider making the switch to developing with Lightning components. Salesforce1 mobile development, for example, is a great place to use Lightning components. Also use Lightning components for new projects and any project involving highly interactive applications.

Okay, so we know what to consider. But where exactly can you *use* Lightning components? You have several options.

Lightning Experience

We said it earlier, but don't want you to get the wrong idea by leaving it off this list. Lightning Experience and Lightning components are two great tastes that taste great together.

Salesforce1

We're repeating this one often because it's important: use Lightning components for your mobile development. When you're using a mobile device, you don't want to make a call to the server every time a user presses a button. Using Lightning components vastly improves mobile app performance.

Standalone Apps

If you used Lightning components in Salesforce Classic, you probably made at least one standalone Lightning app. Lightning App Builder lets you declaratively create apps with standard components ranging from buttons to Canvas apps. Alternatively, use the Developer Console to create apps made up of both standard and custom Lightning components. See the Lightning Components Developer Guide for more information.

Visualforce Pages

This capability is perfect for Salesforce developers who are Visualforce veterans. If you're not quite ready to commit to a full Lightning app, smooth the transition by integrating components into Visualforce pages. This task only requires a few lines of markup and gives you a huge amount of flexibility. See the Lightning Components Developer Guide for more information about Lightning Components for Visualforce.

Anywhere!

Lightning Out, currently available in beta, lets you run your Lightning components and apps, well, pretty much anywhere you can serve a web page. Whether it's a Node.js app running on Heroku, a

> department server inside the firewall, or even SharePoint (yes, SharePoint), build your custom app with Force.com and run it wherever your users are.

As much engineering effort as we've put into making Lightning components a framework you can use to create applications for the next decade, we're not done. There's still a few places where you can use Visualforce to customize Salesforce but you can't yet use Lightning components. Stay tuned to this channel.

CHAPTER 27 ISVs, Packaging, and AppExchange

In this chapter ...

- ISV Tools in Lightning Experience
- Packaging Apps in Lightning Experience
- AppExchange and Lightning Experience

If you're a Salesforce ISV partner, you probably have some concerns about how your app development and release processes are affected by Lightning Experience. Will your existing apps still work? Will you have to make major changes in your existing procedures? How about future app releases?

These are all valid concerns, but we have good news. At the moment, most parts of the ISV experience are the same. There are a few speed bumps to look out for along the way. But overall, we hope that Lightning Experience opens more doors for innovative app development than it closes.

This unit provides a general overview of what ISVs should consider when using Lightning Experience.

ISV Tools in Lightning Experience

First, let's talk tools. This conversation is bittersweet. Some of the tools that you use are available in Lightning Experience, while others aren't supported there yet. Here's an overview of what's supported right now.

Feature	Supported In
• Environment Hub • License Management App (LMA) • Checkout Management App (CMA)	Both Salesforce Classic and Lightning Experience.
• Trialforce • Channel Order App (COA) • Usage Metrics Visualization App	Salesforce Classic only.

This means you can go to Lightning Experience if you want to:

- Create orgs for development, testing, and demos
- Manage all of your orgs from a single location
- License and support apps published on the AppExchange
- View and report on subscription data for apps sold using AppExchange Checkout

To provision trial orgs or manage orders, you should stay in Salesforce Classic. As Lightning Experience continues to mature, more ISV tools will become available.

Packaging Apps in Lightning Experience

When you're ready to distribute an app, Lightning component, or other offering, Lightning Experience is up to the task. The package manager lets you create and manage packages from Setup, just like you could in Salesforce Classic. Whether you're creating managed packages in a Developer Edition or unmanaged packages in an Enterprise Edition, the new packaging experience feels right at home in your development process.

AppExchange and Lightning Experience

If you've already published apps on AppExchange, you probably have concerns about whether they'll work in organizations using Lightning Experience. As you've no doubt noticed, there are still some features that Lightning Experience doesn't support. Depending on the functionality of your apps, there's a chance that they aren't compatible with Lightning Experience.

So what do you do about that? Moving forward, similar to the way that your app undergoes a security review when you list it on AppExchange, it's also reviewed for Lightning Experience readiness. Apps supported in Lightning Experience get a "Lightning Ready" sash on their listing. Apps that aren't certified as Lightning Ready can still be used in Lightning Experience but there's no guarantee that they'll work as expected. They might also be visually inconsistent with Lightning Experience. It's best to use these apps in Salesforce Classic.

The good news here is that, as an ISV, you're not expected to change all your listed apps so that they're Lightning Ready. You can make this transition over time or expect that your users continue using your apps in Salesforce Classic.

CHAPTER 28 Understanding Changes to Other Development Tools

In this chapter ...

- Installed Packages in Lightning Experience
- The API and Apex in Lightning Experience
- Authentication and Security in Lightning Experience
- Canvas for Lightning Experience
- Salesforce1 for Lightning Experience
- Mobile SDK for Lightning Experience

This last unit is a bit of a grab bag. We've already covered all of the "hard" stuff, so at this point you're in the home stretch. Can you feel it? That's the itch you get when you're close to a new badge. Let's do this.

Installed Packages in Lightning Experience

Managing your installed packages in Lightning Experience is the same as it ever was. The Installed Packages landing page is available in the Lightning Experience setup area. It looks and works the same way it does in Salesforce Classic.

Of course, finding and using this page isn't the only thing on your mind. You're also wondering whether your packages installed from AppExchange still work in Lightning Experience.

The best answer we can give is **maybe**. Apps listed on AppExchange are marked with a "Lightning Ready" sash if they're fully compatible with Lightning Experience. Check the listing to see if an app is Lightning Ready. If it's not, you can still try to use it in Lightning Experience, but we recommend sticking to Salesforce Classic to prevent unexpected behavior.

The API and Apex in Lightning Experience

As a developer, one of your most important tools on any platform is the API. As a Salesforce developer, Apex is just as important to your success.

We've kept our promise that we won't do anything that breaks your dependencies on our APIs. Your Apex code and queries continue to function as expected, regardless of whether you're using Lightning Experience or Salesforce Classic. It really is just that simple. Breathe a sigh of relief.

Authentication and Security in Lightning Experience

Regardless of the user experience you're developing for, security is still Salesforce's top priority. Lightning Experience doesn't fall short of our promise to keep your organization's data safe.

Continue to treat authentication and security as you do when developing for Salesforce Classic. The only difference in access controls between Salesforce Classic and Lightning Experience is in the App Launcher. The App Launcher is available by default to all users in your organization. While this change is of concern mostly to administrators, it's important to work with your Salesforce admin to ensure that your development work is only seen by the people who are supposed to see it.

Canvas for Lightning Experience

Force.com Canvas allows you to easily integrate third party applications in Salesforce. Canvas functionality in Lightning Experience is the same as in Salesforce Classic. You can still embed Canvas apps in Visualforce pages, Salesforce1, and everywhere else they're supported—with the added bonus that you can integrate Canvas apps in Lightning Components!

Salesforce1 for Lightning Experience

Lightning Experience and Salesforce1 are like peanut butter and jelly. They're made for each other. Your mobile development practices in Lightning Experience are the same as they were in Salesforce Classic.

When we say the two were made for each other, we mean it. You might be familiar with the `sforce.one` JavaScript object. In the past, it was used as an event mechanism for navigation in Salesforce1 development. Now, you can also use it for navigation in Lightning Experience. See the Salesforce1 Mobile App Developer's Guide for more information.

Mobile SDK for Lightning Experience

By now you're probably tired of reading overview information and are ready to dive in to the nitty-gritty technical details. We won't keep you much longer. Literally just another two sentences!

As its own client, or front end, to Salesforce the Mobile SDK isn't impacted by Lightning Experience. If you use Mobile SDK to develop mobile apps, you can rest easy.

VISUALFORCE & LIGHTNING EXPERIENCE

CHAPTER 29 Using Visualforce in Lightning Experience

In this chapter ...

- Where You Can Use Visualforce in Lightning Experience

Lightning Experience brings an all new user interface to your Salesforce organization, but that doesn't mean your Visualforce apps stop working. Visualforce pages work in Lightning Experience, many without any revisions. Things have moved around, though, and there are some chores you'll want to complete to make sure your Visualforce pages work the way you expect as your users switch between Lightning Experience and Salesforce Classic. And there are a very few features that, alas, don't work in Lightning Experience. We'll get you sorted on all of it in this module.

Let's start with a few basic details. These are topics we'll cover in depth later, but let's handle some essential items right up front.

- With some important exceptions, Visualforce "just works" in Lightning Experience. If you've written Visualforce apps for your organization, you can expect that they work whether your users access them in Lightning Experience or Salesforce Classic.
- If your Visualforce pages use the built-in standard components, their look-and-feel matches Salesforce Classic, whether your users access them in Lightning Experience or Salesforce Classic. If you want your pages to match the Lightning Experience styling, you have some work to do.
- If your Visualforce pages make use of JavaScript, there are things you need to check. Visualforce doesn't "own" the whole page when shown in Lightning Experience, and because of that your JavaScript code needs to play by some new rules.
- There are other things that have changed about how Visualforce runs when it's running inside Lightning Experience. For the most part, these are turning the "just works" crank, but you'll want to be aware of them all the same.

And finally, did we mention that some things have moved around? Have they ever! Lightning Experience is a complete rethinking of how to use Salesforce, and while the job's not done yet, we're really excited about where we're going. To get you oriented for where your Visualforce is in the new environment, let's take a quick tour of some of the places you can use Visualforce in Lightning Experience.

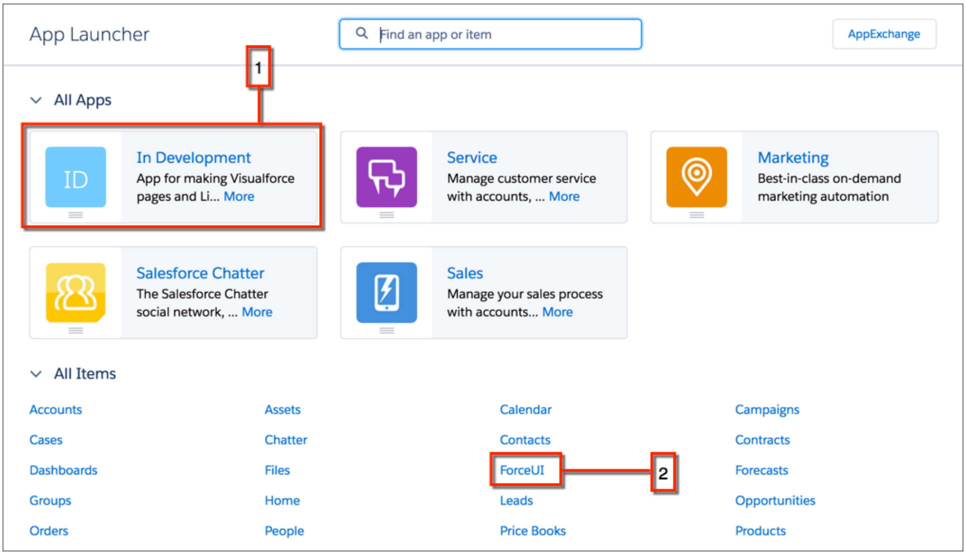
Where You Can Use Visualforce in Lightning Experience

As with Salesforce Classic, you can extend Lightning Experience with your custom Visualforce pages and apps. But where you find them has changed, and there are still some places you can't put Visualforce.

The following are some of the ways you can add Visualforce to your Lightning Experience organization. This is just a quick tour, though. For more details on how to customize your organization using Visualforce pages, see the resources at the end of this unit.

Example: **Open a Visualforce Page from the App Launcher**

Your Visualforce apps and custom tabs are all available from the App Launcher, which you reach by clicking ⋮⋮⋮ in the header.



In Development
App for making Visualforce pages and Li... More

Service
Manage customer service with accounts, ... More

Marketing
Best-in-class on-demand marketing automation

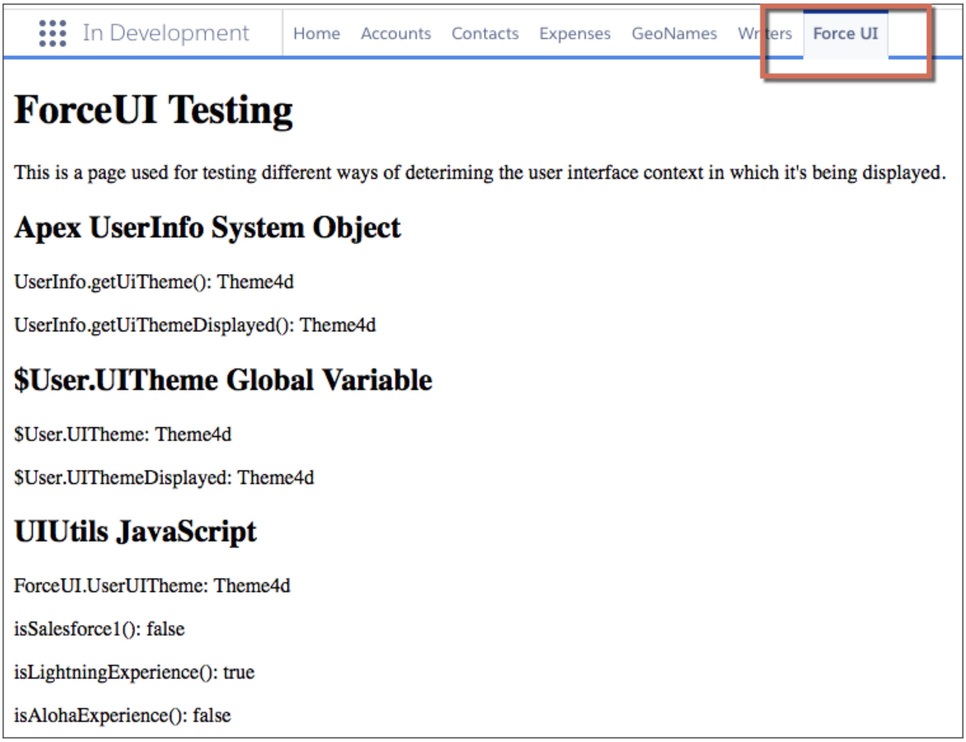
Salesforce Chatter
The Salesforce Chatter social network, ... More

Sales
Manage your sales process with accounts... More

Click a custom app (1) to activate it. Items in the app display in the navigation bar, including any Visualforce tabs you've added to the app. Note that you need to add your Visualforce pages to tabs for them to be accessible in the App Launcher. Visualforce tabs that aren't in apps can be found in All Items (2).

Example: **Add a Visualforce Page to the Navigation Bar**

As described in the preceding example, you can add Visualforce tabs to an app and they display as items in the app's navigation bar.



ForceUI Testing

This is a page used for testing different ways of deteriming the user interface context in which it's being displayed.

Apex UserInfo System Object

UserInfo.getUiTheme(): Theme4d

UserInfo.getUiThemeDisplayed(): Theme4d

$User.UITheme Global Variable

$User.UITheme: Theme4d

$User.UIThemeDisplayed: Theme4d

UIUtils JavaScript

ForceUI.UserUITheme: Theme4d

isSalesforce1(): false

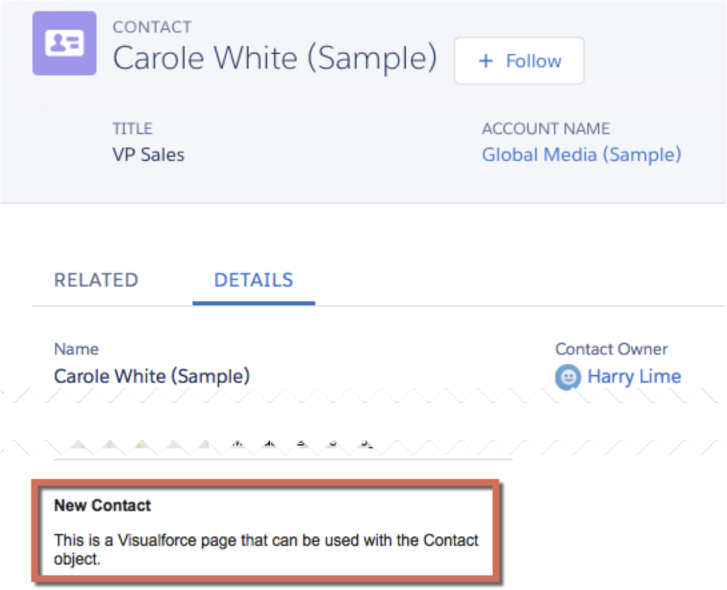
isLightningExperience(): true

isAlohaExperience(): false

(And hey, does a "ForceUI" utility page sound interesting? Keep reading this module!)

Example: Display a Visualforce Page within a Standard Page Layout

Extend your page layouts by embedding Visualforce pages on them to display completely custom content on a standard page. The behavior here is identical to Salesforce Classic, except you need to view the record's Details to see the page layout.

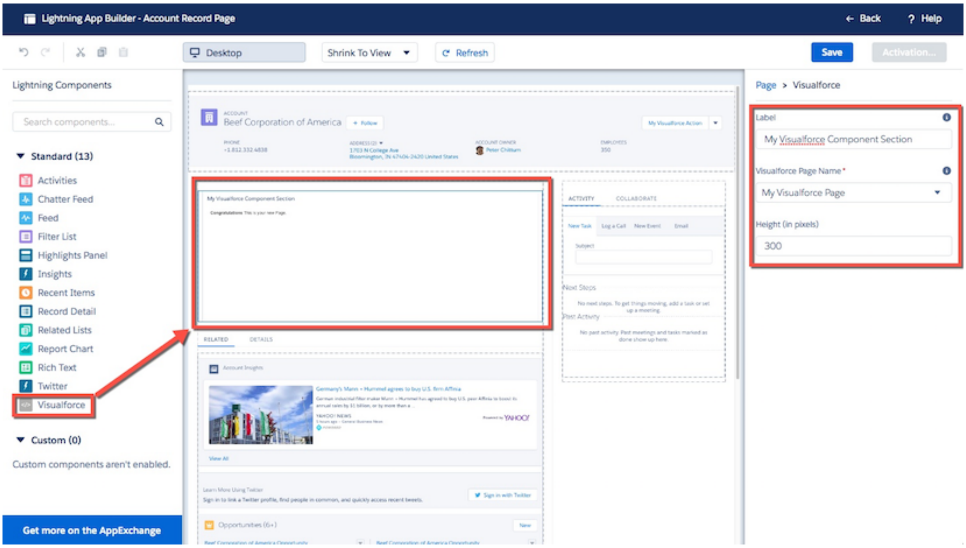


New Contact

This is a Visualforce page that can be used with the Contact object.

Example: Add a Visualforce Page as a Component in Lightning App Builder

When you create a custom app page in Lightning App Builder, you can add a Visualforce page to the page by using the Visualforce component.



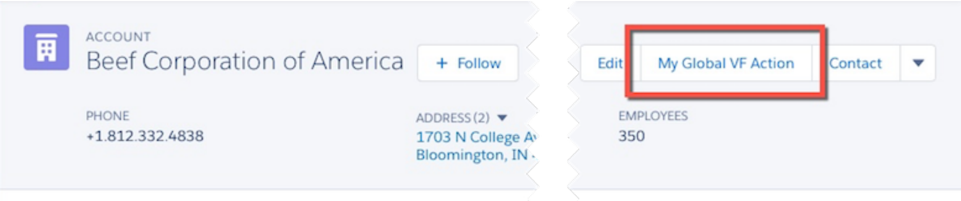
Label: My Visualforce Component Section

Visualforce Page Name*: My Visualforce Page

Height (in pixels): 300

Note: You must enable `Available for Salesforce mobile apps and Lightning Pages` for a Visualforce page to make it available in Lightning App Builder.

Example: Launch a Visualforce Page as a Quick Action

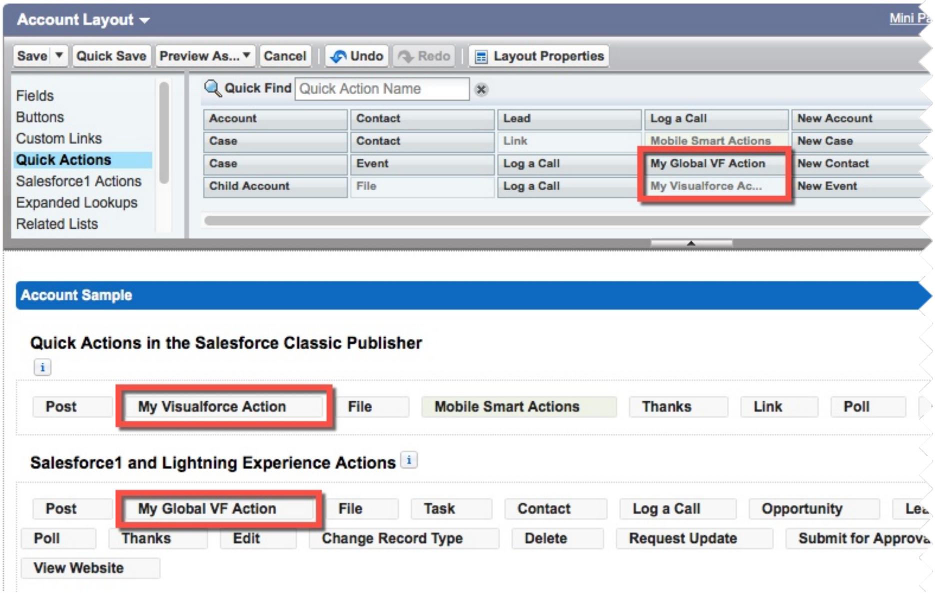


PHONE: +1.812.332.4838

ADDRESS (2): 1703 N College A… Bloomington, IN…

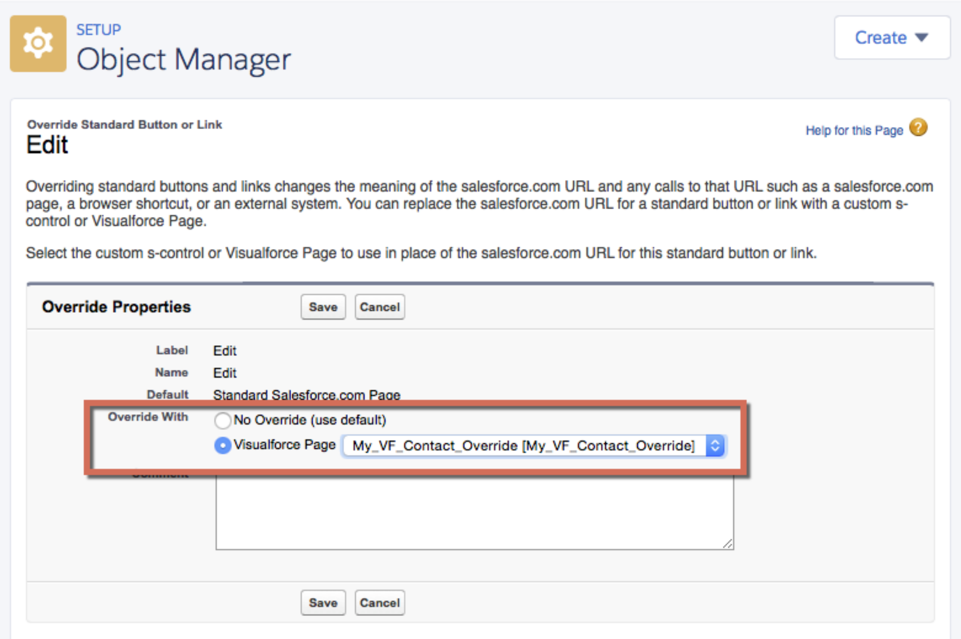
EMPLOYEES: 350

Although their placement in the Lightning Experience user interface is quite different from Salesforce Classic, the process of adding quick actions is much the same. Add them to the appropriate publisher area on the object's page layout.



Example: Display a Visualforce Page by Overriding Standard Buttons or Links

You can override the actions available on an object with a Visualforce page. When the user clicks a button or link that has been overridden, your page displays instead of the standard page. Setting this up is pretty much identical to Salesforce Classic. Indeed, you'll have a hard time telling that you're in Lightning Experience when defining an action override!



Override Standard Button or Link

Edit

Overriding standard buttons and links changes the meaning of the salesforce.com URL and any calls to that URL such as a salesforce.com page, a browser shortcut, or an external system. You can replace the salesforce.com URL for a standard button or link with a custom s-control or Visualforce Page.

Select the custom s-control or Visualforce Page to use in place of the salesforce.com URL for this standard button or link.

Override Properties

Label	Edit
Name	Edit
Default	Standard Salesforce.com Page
Override With	○ No Override (use default) ● Visualforce Page My_VF_Contact_Override [My_VF_Contact_Override]
Comment	

Example: Display a Visualforce Page Using Custom Buttons or Links

You can create new actions for your objects, in the form of buttons and links, by defining them on an object. JavaScript buttons and links aren't supported in Lightning Experience, but Visualforce (and URL) items are. The process of defining Visualforce buttons and links is identical to that in Salesforce Classic, so we won't bother to show it here.

CHAPTER 30 Developing Visualforce Pages for Lightning Experience

In this chapter ...

- Set Up Your Editor
- Viewing Visualforce Pages During Development
- Reviewing Visualforce Pages in Multiple Environments
- Testing Your Visualforce Pages

The development process for creating Visualforce pages and apps for Lightning Experience is in some ways considerably different from developing for Salesforce Classic. In others, you'll find it's just the same. The main difference is how you view and test your pages during development.

In this unit we'll cover the details of getting your development environment set up, and then get into the details of the "right" way to test your pages while you're in the process of building them. The good news is that the process you need to use to develop for Lightning Experience is the same you'll use for developing Salesforce1 pages as well.

Set Up Your Editor

The first thing you'll want to set up is the editing tool you'll use for writing code. This process remains the same, whether you're creating pages for Lightning Experience, Salesforce Classic, or Salesforce1, and whether you're using the Developer Console, the Force.com IDE, or the good old Setup editor.

If you've already got a preferred Visualforce editing tool, you don't need to do anything here. Writing and saving your Visualforce markup remains exactly the same. The Developer Console has its own user interface, and doesn't change between Lightning Experience and Salesforce Classic. The editor in Setup is also unchanged, retaining the Salesforce Classic user interface in all user interface contexts. And of course if you're using a native tool, such as the Force.com IDE or one of the many third-party tools available, those have their own user interfaces.

The one exception is the editor in the Visualforce Development Mode footer. If you've enabled Development Mode in your user settings, and you're using Salesforce Classic, then viewing and editing Visualforce pages with the Development Mode footer is unchanged, as you'd expect. If you switch to Lightning Experience, and then access a page using the traditional `https://`*`yourInstance`*`.salesforce.com/apex/`*`PageName`* URL pattern, you might be somewhat surprised to find yourself back in Salesforce Classic.

This is expected, and we'll talk about it more when we get to viewing and testing your Visualforce pages. For now, know that Development Mode for Visualforce is only available in Salesforce Classic.

Viewing Visualforce Pages During Development

Viewing your Visualforce pages while they're being developed is a common task. And while it's not "testing" in the formal sense, you certainly want to be able to interact with functionality you've built to ensure it's making progress towards correct behavior. This is frequently accomplished by accessing the page using the `https://`*`yourInstance`*`.salesforce.com/apex/`*`PageName`* URL pattern. While this still works for reviewing pages in Salesforce Classic, it doesn't work for checking behavior in Lightning Experience.

Pages you view using direct URL access always display in Salesforce Classic, which is to say, the "classic" Visualforce container, no matter what your user interface settings are. If you create Visualforce pages that have Lightning Experience-specific behavior, you won't be able to review that behavior just by using the usual direct URL access.

Beyond the Basics

What's going on behind the scenes that causes this? It's pretty simple, really. In order to view your page in Lightning Experience, you need to access the Lightning Experience container app. This

> means accessing `/one/one.app`. If you're accessing that, you can't access `/apex/`*`PageName`*. They're just two different URLs that don't overlap.

So what's a developer to do? You need to view your page from within the Lightning Experience app itself, so that it's running inside the Lightning Experience container. This means you'll need to navigate to the page in Lightning Experience, and there's a variety of ways to do that.

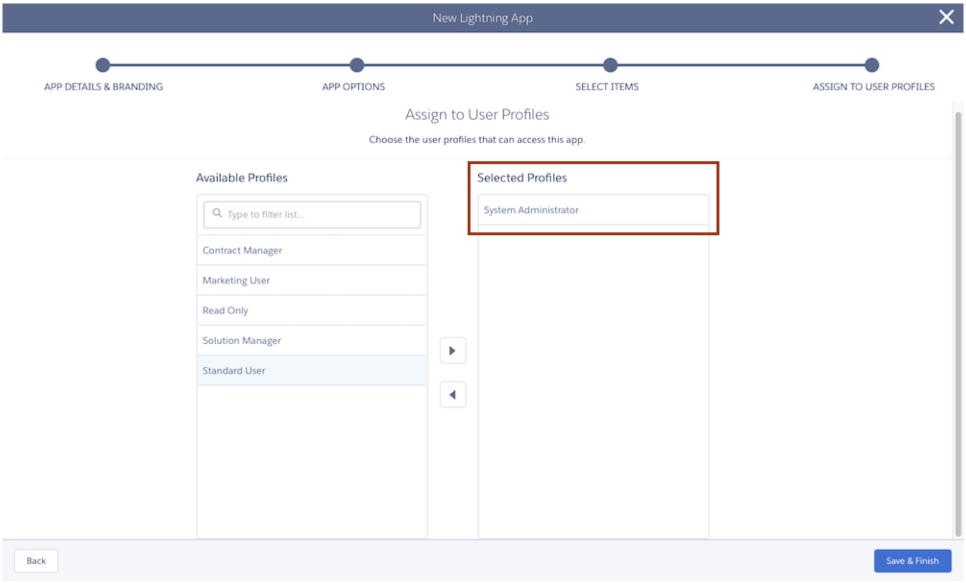
The simplest way to get to a specific Visualforce page is to create a tab for it, and then navigate to that tab via the All Items section in the App Launcher. A more long-term approach would be to create an "In Development" app, and add your Visualforce tabs to it as you work on them, and move or remove them as they roll out to production. Since the controls for doing this have moved around a bit, here are brief instructions.

1. From Setup, enter *`Apps`* in the `Quick Find` box, then select **App Manager**.

   You should see the `Lightning Experience App Manager` Setup page.

2. Click **New Lightning App**, and then create a custom app for your pages in development.

   Consider restricting your app to only System Administrators, or a profile you've created for developers in your organization.



New Lightning App

Assign to User Profiles

Choose the user profiles that can access this app.

Available Profiles

- Contract Manager
- Marketing User
- Read Only
- Solution Manager
- Standard User

Selected Profiles

- System Administrator

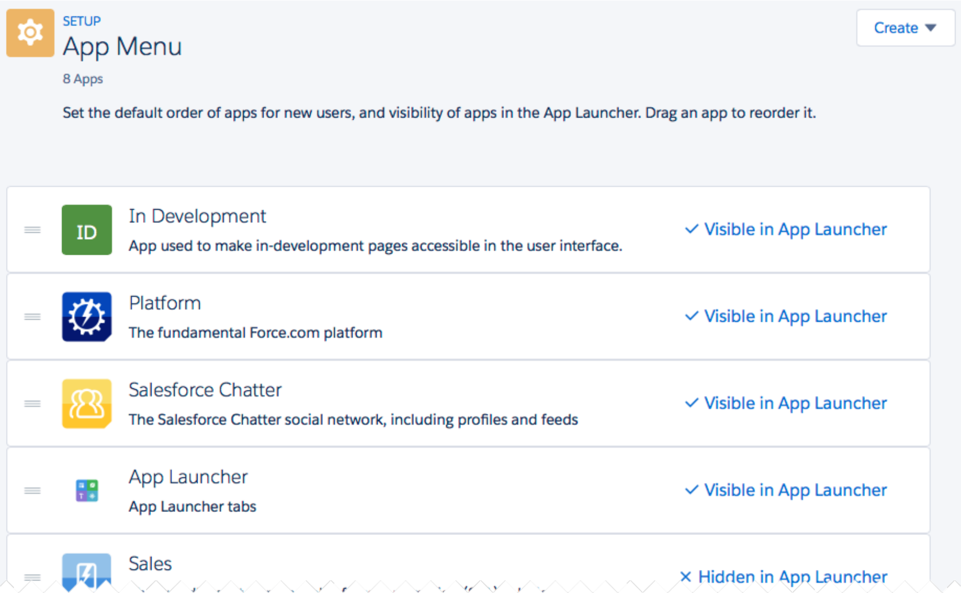
You don't need your users to see your pages before they're added to their permanent place in the user interface.

3. From Setup, enter *`App Menu`* in the `Quick Find` box, then select **App Menu**.

   You should see the `App Menu` Setup page.

4. Make sure your In Development app is set to `Visible in App Launcher`.

   While you're at it, you might want to rearrange items, and even hide apps you don't use.



SETUP

App Menu

8 Apps

Set the default order of apps for new users, and visibility of apps in the App Launcher. Drag an app to reorder it.

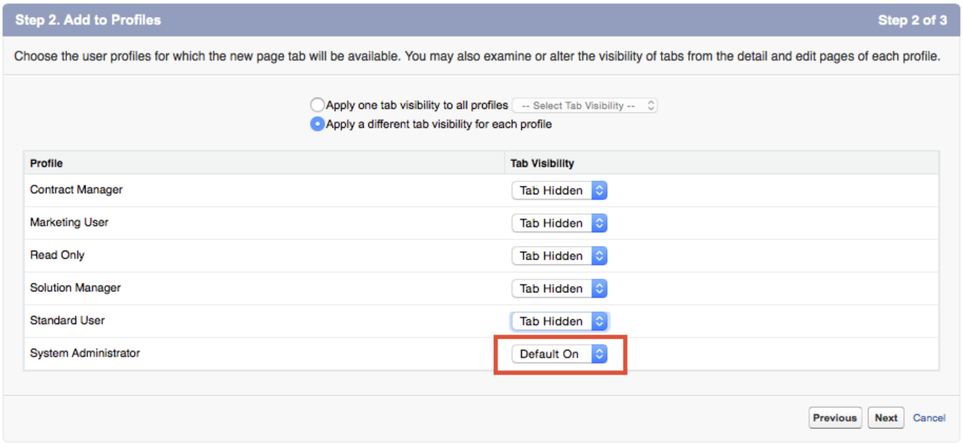
ID	In Development App used to make in-development pages accessible in the user interface.	✓ Visible in App Launcher
	Platform The fundamental Force.com platform	✓ Visible in App Launcher
	Salesforce Chatter The Salesforce Chatter social network, including profiles and feeds	✓ Visible in App Launcher
	App Launcher App Launcher tabs	✓ Visible in App Launcher
	Sales	× Hidden in App Launcher

5. From Setup, enter *`Tabs`* in the `Quick Find` box, then select **Tabs**.

   You should see the `Custom Tabs` Setup page.

6. Click **New** in the Visualforce Tabs section, and then create a custom tab for the page currently in development.

   Make the tab visible only to your development user profile, and add the tab only to your In Development app.

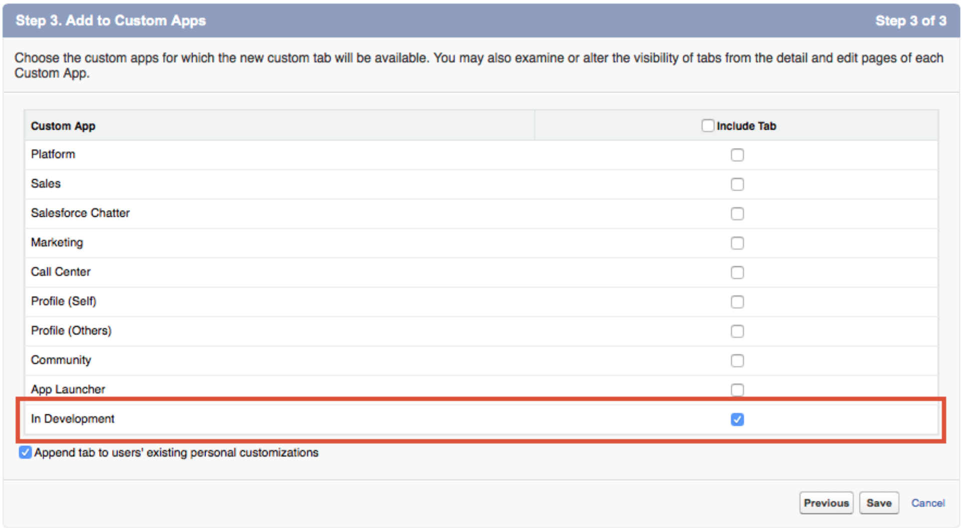


Step 2. Add to Profiles Step 2 of 3

Choose the user profiles for which the new page tab will be available. You may also examine or alter the visibility of tabs from the detail and edit pages of each profile.

○ Apply one tab visibility to all profiles -- Select Tab Visibility --
● Apply a different tab visibility for each profile

Profile	Tab Visibility
Contract Manager	Tab Hidden
Marketing User	Tab Hidden
Read Only	Tab Hidden
Solution Manager	Tab Hidden
Standard User	Tab Hidden
System Administrator	Default On



Step 3. Add to Custom Apps Step 3 of 3

Choose the custom apps for which the new custom tab will be available. You may also examine or alter the visibility of tabs from the detail and edit pages of each Custom App.

Custom App	□ Include Tab
Platform	□
Sales	□
Salesforce Chatter	□
Marketing	□
Call Center	□
Profile (Self)	□
Profile (Others)	□
Community	□
App Launcher	□
In Development	☑

☑ Append tab to users' existing personal customizations

7. Repeat the previous step for each page you want to add to your In Development app. For adding new pages in the future, this is the only step required.

For all that that's an easy way to see your pages while you're working on them, it doesn't really compare to simply typing the page name into a URL. For a similarly low-overhead way to test your page in Lightning Experience, you can type the following into your JavaScript console:

```
$A.get("e.force:navigateToURL").setParams(
    {"url": "/apex/pageName"}).fire();
```

This JavaScript fires the Lightning Experience `navigateToURL` event, and is the equivalent of entering in the classic `/apex/`*`PageName`* URL—you can even see that URL pattern in the code.

Note: You need to currently be in Lightning Experience for this technique to work. If you're in Salesforce Classic, the JavaScript code fails.

For something a little more convenient to use, add the following bookmarklet to your browser's menu or toolbar. (We've wrapped this code for readability.)

```
javascript:(function(){
    var pageName = prompt('Visualforce page name:');
    $A.get("e.force:navigateToURL").setParams(
        {"url": "/apex/" + pageName}).fire();})();
```

This bookmarklet prompts you for a page name, and then fires the event to navigate directly to it. Useful!

Once you've navigated to the page you're working on, you can simply use your browser's reload command to refresh the page as you make changes.

Reviewing Visualforce Pages in Multiple Environments

If you're creating pages that will be used in Lightning Experience, Salesforce Classic, and Salesforce1, you'll want to be able to review them in all environments while you're working on them. To do so, you'll need to open the page in multiple browsers and on multiple devices.

A Visualforce page that's going to be used across the different Salesforce user interface contexts and form factors is tricky to review while you're in development. You can toggle back and forth between Salesforce Classic and Lightning Experience using the environment selector in the profile menu, but that's going to get old quickly. And you can similarly play with your browser's user agent settings to simulate the Salesforce1 environment, but that's even more cumbersome.

Instead, you're going to want to use multiple browsers, or even multiple devices, to view your pages. And you'll want to have access to at least one additional test *user* as well. Here's an example of how you might set up your development environment.

Main Development Environment

This environment is where you work in Setup to make changes to your organization, like adding custom objects and fields, and maybe where you write actual code, if you use the Developer Console.

- **Browser**: Chrome
- **User**: Your developer user
- **User interface setting**: Salesforce Classic

Review your page's design and behavior in Salesforce Classic in this environment.

Lightning Experience Review Environment

This environment is where you check your page's design and behavior in Lightning Experience.

- **Browser**: Safari or Firefox
- **User**: Your test user
- **User interface setting**: Lightning Experience

Salesforce1 Review Environment

This environment is for checking your page's design and behavior in Salesforce1.

- **Device**: iOS or Android phone or tablet
- **Browser**: Salesforce1 app
- **User**: Your test user
- **User interface setting**: Lightning Experience

Note: This is just an example setup, and you can use any modern browsers or mobile devices in yours, for both Salesforce Classic and Lightning Experience. The key is to use two different browsers so you can access both Salesforce Classic and Lightning Experience at once, and to use a real device to test with Salesforce1.

This probably sounds pretty elaborate, and it *is* a bit of a chore to set up initially, especially if you're champing at the bit to get coding. But keep in mind two things. First, once you're set up, it's done. And second, this workspace doesn't just give you a great development environment, it also provides you with the environments you need for formal testing of your pages. Because you wouldn't dream of putting your pages into production without formal testing, right?

Testing Your Visualforce Pages

Testing your Visualforce pages before deploying them into production is an essential development task. When your organization adopts Lightning Experience the process of testing your pages becomes more complicated.

We just talked about the environments you need to set up for doing quick, informal testing while you're developing your pages. The need for those environments also applies to testing your pages and apps formally. Rather than repeating them, let's talk a bit about *why* you need these different environments.

The need to test your pages in both Lightning Experience and Salesforce Classic is fairly obvious, but why can't you do that testing in the same browser, with the same user? In fact, you can and you should. Your users can toggle back and forth between the different user interfaces, and you should verify that your pages work when they do so.

But you also want to test pages in a more isolated and systematic fashion, so that you're sure that what you're testing is the page's basic functionality, as separated as possible from the effects of other code, whether that code is yours, ours, the browser's, or the device's.

This highlights another testing issue. In the past we've suggested that it's reasonable to do Salesforce1 development on your desktop or laptop, and navigating to the `/one/one.app` URL used by the Salesforce1 app. This method doesn't work anymore, because `/one/one.app` is shared by both Salesforce1 and Lightning Experience, depending on the device that connects to it. While you can fool `/one/one.app` by changing your browser's user agent, this is a bad idea, an anti-pattern. The reason is that desktop and mobile browsers, even from the same vendor, behave differently—sometimes *very* differently. You cannot do rigorous, formal testing unless you're testing on every device and every browser you plan to support.

If you're developing an individual page, or a basic app for identical devices, your "matrix" of the different factors might be simple. But for more ambitious projects, if you're developing functionality that you need to support across a range of possibilities, your test plan should take into consideration the need to test across:

- Each different supported device.
- Each different supported operating system.
- Each different supported browser—including the Salesforce1 app, which embeds its own.
- Each different supported user interface context (Lightning Experience, Salesforce Classic, and Salesforce1).

Fortunately for your sanity, some of these factors collapse together, reducing the combinatorial explosion. For example, most Apple mobile devices can be counted on to be updated to the latest version of iOS. This means that the device, operating system, and browser are effectively only one combination. Your test plan might therefore choose to test only one iPhone and one iPad, updated to the latest iOS and Salesforce1 app.

A final word about testing. Another reason we strongly suggest your development and test environments be similar is so that you can start testing, full testing, early in the development process. We've found that it's all too easy to put testing on secondary devices off until late in a project. When that happens it's inevitably a setback when problems are discovered.

Test early, test often, test everything.

CHAPTER 31 Exploring the Visualforce App Container

In this chapter ...

- The Outer Lightning Experience Container
- The Visualforce iframe
- Impact of the New Container
- Visualforce Defaults and Environment Changes in Lightning Experience
- <apex:page> showHeader and showSidebar Attributes Are Always false
- The sforce.one JavaScript Utility Object

The largest difference between Visualforce in Lightning Experience and Visualforce in Salesforce Classic is the environment it runs in. In Salesforce Classic, Visualforce “owns” the page, the request, the environment. Visualforce *is* the application container. But in Lightning Experience, Visualforce runs inside an iframe that’s wrapped inside the larger Lightning Experience container.

This change to the execution context has a number of effects on the way Visualforce pages can affect the overall Salesforce application. We’ll talk about these changes in this unit, but save the full details of a few of them for their own units.

Note: This unit is a little more “under construction” than the rest. The reason is simple: The impact of the issues described here is highly dependent on your code. We’ve worked really hard to make things “just work” for you, and in most cases little or nothing here will show on your radar. But we can’t anticipate every way that you’re using Visualforce. Here we’re outlining the general aspects of how Lightning Experience affects Visualforce. When you have conversations with us, and as we learn more from you about actual impact, we can offer explanations with more details about how to address specific issues.

The Outer Lightning Experience Container

Let's start with the outer container, the Lightning Experience application. The Lightning Experience container is a "single-page application," or SPA, which is accessed at the `/one/one.app` URL. The `one.app` page loads, its code starts up, and that application code takes over the environment.

The process by which a single-page application loads its resources—usually a static HTML shell and a lot of JavaScript—is both interesting and complex. If you've worked with JavaScript frameworks like AngularJS or React, you're reasonably familiar with the basics of how Lightning Experience, in the form of `one.app`, starts up. And to be honest, the full details don't matter. You don't have any control over it, and the implementation continues to evolve.

Here's what's important to know: Lightning Experience, or `one.app`, is in charge of the request. Your Visualforce page is not. Your page needs to work within constraints that Lightning Experience imposes upon it. Lightning Experience is the parent context, and your Visualforce page is the child context. Children need to obey their parents.

Some of these constraints, such as the size of the frame in which your Visualforce page is displayed, are imposed directly by Lightning Experience. They're easier to understand and work with, and we'll talk about them in a minute.

Other constraints are implicit, and enforced not by Lightning Experience but by the browser running it. These are mostly security and JavaScript execution constraints. Most pages aren't impacted by these security constraints, and those that are usually fail early and with clear error messages. JavaScript errors are harder to discover and diagnose, but there are some general rules we'll cover in a bit.

The Visualforce iframe

When your Visualforce page runs in Lightning Experience, it's displayed inside an HTML iframe. An iframe creates an embedded browsing context that's effectively a separate browser "window" from the main Lightning Experience browsing context. The iframe creates a boundary between the Visualforce page and its parent, the Lightning Experience application.

The advantage of running Visualforce pages inside an iframe is that, for pages that don't need to access or change the top-level browsing context, running inside the iframe looks almost exactly like running as a page in Salesforce Classic. This is why you don't need to modify all of your Visualforce pages to adapt to the wildly different behind-the-scenes request environment of Lightning Experience. It's an important part of the "just works" strategy for supporting Visualforce.

Of course, the flip side is that pages that *do* need to access the top-level browsing context, well, there's some things that need to change. We'll cover some specifics in the next section.

If your page is communicating with services besides Salesforce, the iframe boundary might also result in you needing to update your organization's CORS settings, remote site settings, clickjack settings, or content security policy. Since these depend on security policies and settings outside of Salesforce, we can't provide a recipe for specific changes. We simply call it to your attention here.

Impact of the New Container

The effects of the new Visualforce container—embedding the Visualforce page into an iframe within the Lightning Experience app—can be broadly divided into two categories, which we'll call *security* and *scope*.

Again, we want to emphasize: many, or even most Visualforce pages won't be affected by these issues. But for those that are, we're thinking "forewarned is forearmed." You'll find the source of the problem faster if we've already talked about it together.

Security Impact

Elements of security that might be affected include the following.

- Session maintenance and renewal
- Authentication
- Cross-domain requests
- Embedding restrictions

We discussed a few of these briefly already, the items dealing with cross-domain requests. That is, when the content in the full browser window comes from requests to different servers and services, there's the potential for any of those requests to balk at being displayed in a context that it's not prepared for. Your mission, should the need arise, is to prepare those services to handle requests intended to be put together within the Lightning Experience context. As we said before, the details vary, so we can't provide specific answers here.

One thing we do want to mention specifically is session maintenance. A "session" for our purposes here is basically some kind of token that your browser re-uses from request to request so that you don't need to enter your username and password for every request. You often need to access the current session using the global variable `$Api.Session_ID`.

Here's the thing to keep in mind. `$Api.Session_ID` returns different values depending on the domain of the request. This is because the session ID varies during a session whenever you cross a hostname boundary, such as `.salesforce.com` to `.visual.force.com`. Normally Salesforce transparently handles session hand-off between domains, but if you're passing the session ID around yourself, be aware that you might need to re-access `$Api.Session_ID` from the right domain to ensure a valid session ID.

Lightning Experience and Visualforce pages are not only held in different browser contexts, they're also served from different domains. So, even though it's all showing in one browser window, the session ID *inside* the Visualforce iframe will be different than the session ID *outside* the iframe, in another part of Lightning Experience. Salesforce and Lightning Experience handle this transparently in normal use. But if you're passing around the session ID like hors d'oeuvre at a party (not usually a good idea), you might need to review how you're handling it.

Scope Impact

When we talk about scope we're mainly talking about the following kinds of things.

- DOM access and modification
- JavaScript scope, visibility, and access
- JavaScript global variables such as `window.location`

If this list sounds complicated or confusing, don't worry, we can boil it down to something simple and easy to remember: Don't touch someone else's stuff. Specifically, your JavaScript code (and stylesheet rules, for that matter) can affect elements—DOM nodes, JavaScript variables, and so on—in your page's browsing context, but it can't access elements in any other browsing context, like the parent Lightning Experience context. Don't touch other contexts' stuff!

Practically speaking, the most common code pattern where you'd want to do this kind of thing is to manipulate `window.location` to navigate to another page. This is such a common thing to do, we've written up details on this specific issue...well, by the time you're done with this module, you'll be sick of hearing about it, we promise.

One last note. If you're an experienced JavaScript developer, you're probably already thinking you know how to deal with "I don't have access to the parent browsing context" issues, by using `contentWindow`, `window.parent`, or the like. Please don't. You'll likely run afoul of the same-origin policy (Visualforce and Lightning Experience are served from different domains, remember?). Even if you don't, you're probably replacing obvious, blocking bugs with subtle, intermittent bugs. Where do you want to spend your time: Doing things right, or the debugger?

Doing things right means calling APIs we've made available in your Visualforce pages, primarily for navigation. If you really need to affect things across frame boundaries, use `window.postMessage` to send a message to receiving code in the other frame.

Visualforce Defaults and Environment Changes in Lightning Experience

When your Visualforce pages run in Lightning Experience a number of low-level changes happen behind the scenes. These changes enable most pages to "just work" in the Lightning Experience container, and sometimes you can just be happy they're there. But you'll still want to know they're happening, especially when you're working on advanced application flows, or troubleshooting a tricky problem.

Some of these changes are simple, and obvious once you think about them. For example, Visualforce pages that run in Lightning Experience always have the standard Salesforce Classic header and sidebar suppressed. Other changes aren't as visible, but have just as large an impact.

`<apex:page>` `showHeader` and `showSidebar` Attributes Are Always `false`

These attributes affect the Salesforce Classic header and sidebar on Visualforce pages. The Salesforce Classic header and sidebar are always suppressed when pages run in Lightning Experience, in favor of Lightning Experience navigation elements. There are no corresponding attributes to affect the Lightning Experience header or sidebar because they can't be suppressed.

If your page is shared between Salesforce Classic and Lightning Experience, you can still set these attributes to the values you'd like to use when the page runs in Salesforce Classic.

Note: The `standardStylesheets` attribute of `<apex:page>`, which determines whether to include or suppress the standard Salesforce Classic stylesheets, is unaffected by Lightning Experience. That is, it defaults to `true` in Lightning Experience, but you're able to change it.

The `sforce.one` JavaScript Utility Object

Although `sforce.one` sounds like a droid working in the Salesforce cantina,* it's actually a utility object that provides a number of useful functions you can use in your own JavaScript code.

`sforce.one` is automatically injected into your page when it runs in Lightning Experience or Salesforce1. You'll see it in your JavaScript debugger console and web developer resources list. There's nothing you need to do to add it, and there's no way to suppress it, either. (Sadly, there's no way to get `sforce.one` in your Visualforce pages in Salesforce Classic.)

`sforce.one` is primarily used to fire navigation events. The full details are in an upcoming unit, Managing Navigation.

[*] There is no Salesforce cantina. Alas.

CHAPTER 32 Sharing Visualforce Pages Between Classic and Lightning Experience

In this chapter ...

- Detecting and Responding to the User Experience Context in Visualforce Markup
- Detecting and Responding to the User Experience Context in JavaScript
- Determining the User Experience Context in Apex
- Querying for Lightning Experience via SOQL and API Access

We recommend that, wherever possible, you create Visualforce pages that behave correctly whether they run in Salesforce Classic or Lightning Experience. The benefits in terms of reduced complexity in your organization's code and configuration are obvious. And there are a number of contexts, such as Visualforce overrides of standard actions, where you don't have a choice. An action override always uses the same page, whether you're running in Salesforce Classic, Lightning Experience, or Salesforce1.

It's perfectly reasonable, though, to want slightly or significantly different behavior or styling that's based on the user experience context in which the page is running. In this unit we'll look at a variety of ways to create pages that work correctly in all user experiences, and how your code can detect and make changes for specific contexts.

Detecting and Responding to the User Experience Context in Visualforce Markup

Use the `$User.UITheme` and `$User.UIThemeDisplayed` global variables to determine the current user experience context. You can use these variables in Visualforce expressions to adapt your pages to Lightning Experience, Salesforce Classic, and Salesforce1.

These global variables return a string that uniquely identifies the current user interface context. The possible values for `$User.UITheme` and `$User.UIThemeDisplayed` are the same:

- `Theme1`—Obsolete Salesforce theme
- `Theme2`—Salesforce Classic 2005 user interface theme
- `Theme3`—Salesforce Classic 2010 user interface theme
- `Theme4d`—Modern "Lightning Experience" Salesforce theme
- `Theme4t`—Salesforce1 mobile Salesforce theme
- `PortalDefault`—Salesforce Customer Portal theme
- `Webstore`—Salesforce AppExchange theme

The difference between the two variables is that `$User.UITheme` returns the look and feel the user is *supposed* to see, while `$User.UIThemeDisplayed` returns the look and feel the user *actually* sees. For example, a user may have the preference and permissions to see the Lightning Experience look and feel, but if they are using a browser that doesn't support that look and feel, for example, older versions of Internet Explorer, `$User.UIThemeDisplayed` returns a different value. In general, your code should use `$User.UIThemeDisplayed`.

The simplest way to use these theme globals is to use one in a Boolean expression, like `{! $User.UIThemeDisplayed == "Theme3" }`, in the `rendered` attribute of a component. The component will only display if the page appears in the desired user interface context.

```
<apex:outputText value="This is Salesforce Classic."
    rendered="{! $User.UIThemeDisplayed == 'Theme3' }"/>
```

Although you can use this technique on individual user interface elements, it's usually more efficient if you wrap larger chunks of markup into an `<apex:outputPanel>` or similar block-level component, and then create separate blocks for each different UI you want to present. Then place the theme test on the `rendered` attribute of the blocks, rather than the individual components. Not only should this perform better, your code will be less complicated.

```
<apex:outputPanel rendered="{! $User.UIThemeDisplayed == 'Theme3' }">

    <apex:outputText value="This is Salesforce Classic."/>
    <apex:outputText value="These are multiple components wrapped by 
an outputPanel."/>
</apex:outputPanel>
<apex:outputPanel rendered="{! $User.UIThemeDisplayed == 'Theme4d' }">

    <apex:outputText value="Everything is simpler in Lightning 
Experience."/>
</apex:outputPanel>
```

Another strategy you can use this with is to dynamically select a stylesheet to include on your page, and provide a different stylesheet for each theme. This is a bit trickier than you might think, because the `<apex:stylesheet>` tag doesn't have a `rendered` attribute of its own. In this case, you must wrap the stylesheet components within another component that does have a `rendered` attribute. Here's an example of how to provide a different stylesheet for each of the three modern themes supported by Salesforce.

```
<apex:page standardController="Account">

    <!-- Salesforce Classic "Aloha" theme -->
    <apex:variable var="uiTheme" value="classic2010Theme"
        rendered="{!$User.UIThemeDisplayed == 'Theme3'}">
        <apex:stylesheet value="{!URLFOR($Resource.AppStyles, 
                                         'classic-styling.css')}" />
    </apex:variable>

    <!-- Lightning Desktop theme -->
    <apex:variable var="uiTheme" value="lightningDesktop"
        rendered="{!$User.UIThemeDisplayed == 'Theme4d'}">
        <apex:stylesheet value="{!URLFOR($Resource.AppStyles, 
                                         'lightning-styling.css')}" 
/>
    </apex:variable>

    <!-- Salesforce1 mobile theme -->
    <apex:variable var="uiTheme" value="Salesforce1"
        rendered="{!$User.UIThemeDisplayed == 'Theme4t'}">
        <apex:stylesheet value="{!URLFOR($Resource.AppStyles, 
                                         'mobile-styling.css')}" />
    </apex:variable>

    <!-- Rest of your page -->

    <p>
```

```
        Value of $User.UIThemeDisplayed: {! $User.UIThemeDisplayed }
    </p>
</apex:page>
```

Beyond the Basics

This is an unusual way to use `<apex:variable>`, because we're not actually interested in the value of the variable created. Instead we just want a component that doesn't render any output of its own to wrap the `<apex:stylesheet>` component. You can think of this as `<apex:variable>` "lending" its `rendered` attribute to the wrapped `<apex:stylesheet>` component.

It's a good thing we don't care about the variable itself, because another unusual aspect of wrapping the `<apex:variable>` component around something else is that the variable isn't actually created! Feature or bug? Let's call it...undefined behavior, and avoid using the `uiTheme` variable elsewhere.

Detecting and Responding to the User Experience Context in JavaScript

Detecting the current user experience context in JavaScript code is important if you're using JavaScript heavily in your pages and apps. It's especially important for using the right technique to manage navigation in your JavaScript code. The best way to handle UX context detection in your JavaScript code is with a library of utility functions you can use everywhere.

At first this seems simple, just test against the same global variables available in Visualforce markup. Maybe something like this.

```
function isLightningDesktop() {
    return( "{! $User.UIThemeDisplayed }" == "Theme4d");
}
```

And if you add this code to a Visualforce page, lo, it works.

Here's the problem. As soon as you move this code into a static resource—a best practice for code organization, improving performance, and other reasons—it stops working, because global variables aren't available in static resources. Static resources aren't processed for markup or expressions or globals—or at all. They're simply served up. It's why we call them *static* resources. ;-)

So how do we make this work, without putting duplicate JavaScript code in every page that needs to test for the user experience context? By creating a very simple page that does nothing but inject the

$User.UIThemeDisplayed value into the right JavaScript context, and then using <apex:include> to add it to a page. Then we can test against the injected value in our actual utility code.

Here's the "shim" Visualforce page that we use to inject the $User.UIThemeDisplayed global variable into a JavaScript context, as well as include the JavaScript utility static resource that uses it.

```
<apex:page docType="html-5.0" applyBodyTag="false" applyHtmlTag="false"

           showHeader="false" standardStylesheets="false">

<!-- UIUTILS SCRIPT -->
<apex:includeScript value="{!URLFOR($Resource.ForceUI)}"/>
<!-- UIUTILS SCRIPT -->

<!-- UITHEME INJECTOR -->
<script type="text/javascript">
    (function(myContext){
        // Don't overwrite ourself if we already exist.
        myContext.ForceUI = myContext.ForceUI || {};

        // Because this is Visualforce, not a static resource,
        // we can access a global variable in an expression.
       myContext.ForceUI.UserUITheme = '{! $User.UIThemeDisplayed }';

    })(this);
</script>
<!-- UITHEME INJECTOR -->

</apex:page>
```

This page does two things. First, it pulls in the JavaScript static resource that contains the actual utility method code. (Which, I promise, we *will* get to.) Second, it has an inline JavaScript that, because it's running in a Visualforce page instead of a static resource, can evaluate an expression with the theme global. This script sets a variable inside the ForceUI utility object. This copies the theme value from Visualforce into JavaScript, so that it can be referenced by the JavaScript code that's in the static resource.

This "page" isn't intended to be accessed directly, but rather included into your real pages. This makes the work of adding the JavaScript utility methods a single line in those pages. (Raise your hand if you remember when #include wasn't something you did on social media. Right there with you. Now let's get those kids off that lawn!)

Let's look at how this is used. Here's a very simple page that shows how to add the JavaScript utility methods to the page, using the <apex:include> component within the <head> block. At the bottom of the page is a short bit of JavaScript that illustrates how to use the utility methods from within JavaScript. We've added emphasis to the code to highlight these elements.

```
<apex:page standardController="Account" extensions="ForceUIExtension"

           showHeader="false" standardStylesheets="false"
           applyHtmlTag="false" applyBodyTag="false"
           docType="html-5.0" title="ForceUI Utilities">

<html lang="en">
  <head>
    <title>ForceUI Utilities</title>
    <apex:include pageName="UIThemeUtilsInclude"/>
  </head>

  <body>

    <h1>ForceUI Utilities</h1>

    <p>This is a page used for testing different ways of determining

       the user interface context in which it's being displayed.</p>

    <h2>$User.UITheme Global Variable</h2>

    <p><label>$User.UITheme</label>: {! $User.UITheme }</p>
   <p><label>$User.UIThemeDisplayed</label>: {! $User.UIThemeDisplayed
 }</p>


    <h2>UIUtils JavaScript</h2>

    <p><label>ForceUI.UserUITheme</label>:
       <span id="UserUIThemeJS">(loading...)</span></p>

    <p><label>isSalesforce1()</label>:
       <span id="isSalesforce1JS">(loading...)</span></p>

    <p><label>isLightningExperience()</label>:
       <span id="isLightningExperienceJS">(loading...)</span></p>

    <p><label>isSalesforceClassic()</label>:
       <span id="isSalesforceClassicJS">(loading...)</span></p>
    <script type="text/javascript">
      document.addEventListener('DOMContentLoaded', function(event){
          // Diagnostic only -- don't use this value directly
          document.getElementById('UserUIThemeJS').innerHTML =
ForceUI.UserUITheme;
          // Instead, use these utility methods
          document.getElementById('isSalesforce1JS').innerHTML =
              ForceUI.isSalesforce1();
         document.getElementById('isLightningExperienceJS').innerHTML
 =
              ForceUI.isLightningExperience();
          document.getElementById('isSalesforceClassicJS').innerHTML
=
              ForceUI.isSalesforceClassic();
      });
    </script>
  </body>
</html>
</apex:page>
```

View this page in your org, in Lightning Experience, Salesforce Classic, and even Salesforce1 to confirm that the values changes depending on the environment.

Finally (`#finally`), here's the utility library that contains the JavaScript utility functions that let you create if expressions to conditionally affect the results of your app's JavaScript code based on the user interface context it's running in.

```
// This is an anonymous self-executing function closure thingie,
// like all the cool kids are using these days.
(function(myContext){

    // Deal with possible order-of-execution issues.
    // Don't overwrite ourself if we already exist.
    myContext.ForceUI = myContext.ForceUI || {};

    // Utility methods that make simple string comparisons
    // against a local UserUITheme value. This value is
    // injected from a Visualforce page to allow expression
    // evaluation of the $User.UIThemeDisplayed global.
    myContext.ForceUI.isSalesforceClassic = function() {
        return (this.UserUITheme == 'Theme3');
    }
    myContext.ForceUI.isLightningExperience = function() {
        return (this.UserUITheme == 'Theme4d');
    }
    myContext.ForceUI.isSalesforce1 = function() {
        return (this.UserUITheme == 'Theme4t');
    }
})(this);
```

Except for the possibly unfamiliar syntax of the self-executing function, the code here is laughably simple. The result of the code executing is a utility object, `ForceUI`, added to the global scope of your page. The object receives, from the injector JavaScript in the earlier Visualforce shim page, the value of the `$User.UIThemeDisplayed` global variable. This value is saved in a local variable named `UserUITheme`, which you should treat as a private implementation detail. Never access it directly!

The public API of the object is exposed as a series of functions, `isLightningExperience()` and so on, which you use in the rest of your code as illustrated above. You can even add your own additional functions, for example, to tell mobile from desktop, or `one.app` from plain Visualforce.

Determining the User Experience Context in Apex

Use the `UserInfo.getUiTheme()` and `UserInfo.getUiThemeDisplayed()` system methods to determine the current user experience context in Apex code. You can use them when your controller action methods or properties need to behave differently in different contexts.

The following example illustrates how to use these methods by making them available via getter methods in a controller extension.

```
public with sharing class ForceUIExtension {

   // Empty constructor, required for Visualforce controller extension

    public ForceUIExtension(ApexPages.StandardController controller)
{ }

    // Simple accessors for the System.UserInfo theme methods
    public String getContextUserUiTheme() {
        return UserInfo.getUiTheme();
    }
    public String getContextUserUiThemeDisplayed() {
        return UserInfo.getUiThemeDisplayed();
    }

}
```

You could of course work with the values in your Apex code, rather than directly returning the method call results.

These Apex system methods return a string that uniquely identifies the current user interface context. The possible values returned by these methods are the same as those returned by the `$User.UITheme` and `$User.UIThemeDisplayed` global variables.

- `Theme1`—Obsolete Salesforce theme
- `Theme2`—Salesforce Classic 2005 user interface theme
- `Theme3`—Salesforce Classic 2010 user interface theme
- `Theme4d`—Modern “Lightning Experience” Salesforce theme
- `Theme4t`—Salesforce1 mobile Salesforce theme
- `PortalDefault`—Salesforce Customer Portal theme
- `Webstore`—Salesforce AppExchange theme

Using these methods in server-side controller code should be rare, at least compared to providing different Visualforce markup or JavaScript code. It’s a best practice for your controller and controller extension code to be neutral in terms of the UX context. Let your front end code, whether Visualforce or JavaScript, handle the user interface differences.

Querying for Lightning Experience via SOQL and API Access

Although we don’t recommend this technique, you can query for the current user’s preferred user experience directly using SOQL.

The basic SOQL query is the following.

```
SELECT UserPreferencesLightningExperiencePreferred FROM User WHERE Id
 = 'CurrentUserId'
```

The result is a raw preference value, which you need to convert into something useable.

Here’s just about the simplest possible Visualforce page that runs the above SOQL query and displays the result on the page.

```
<apex:page>

<script src="/soap/ajax/36.0/connection.js"
type="text/javascript"></script>
<script type="text/javascript">
```

```
    // Query for the preference value
    sforce.connection.sessionId = '{! $Api.Session_ID }';
    var uiPrefQuery = "SELECT Id,
UserPreferencesLightningExperiencePreferred " +
                      "FROM User WHERE Id = '{! $User.Id }'";
    var userThemePreferenceResult =
sforce.connection.query(uiPrefQuery);

    // Display the returned result on the page
    document.addEventListener('DOMContentLoaded', function(event){
       document.getElementById('userThemePreferenceResult').innerHTML
 =
            userThemePreferenceResult;
    });
</script>

<h1>userThemePreferenceResult (JSON)</h1>

<pre><span id="userThemePreferenceResult"/></pre>

</apex:page>
```

Querying for the user’s Lightning Experience preference directly is discouraged. The result tells you what the user’s current preference *setting* is, not what user experience actually is on their screen. There are several use cases where the preference value might not reflect the user experience that’s actually being delivered. To determine the actual user experience being delivered in the current request, use `$User.UIThemeDisplayed` or `UserInfo.getUiThemeDisplayed()`.

CHAPTER 33 Managing Navigation

In this chapter ...

- Navigation in Lightning Experience
- Classic Visualforce Navigation
- Modern Visualforce Navigation
- Navigation Gotchas, and How to Fix Them

App flow and navigation is in many ways the heart of application design. Visualforce provides a number of ways to add navigation elements and to direct application flow. Lightning Experience adds its own application flow, navigation elements, and mechanisms for affecting where users go as they use Salesforce.

The good news is that "classic" Visualforce navigation continues to work. The better news is that your Visualforce pages can take advantage of the new Lightning Experience mechanisms, too.

Navigation in Lightning Experience

Before we talk about the details of Visualforce navigation, and how you create it so that it works in Salesforce Classic and Lightning Experience, let's talk a little about navigation in general. What do we actually mean by "navigation"?

The first thing we might mean by navigation is user interface elements on the screen. You click something, and something happens. For example, you click the Accounts item in the navigation menu, and you go to the Accounts object home page. You click the New button, and a record entry form appears. You choose a custom action from a quick actions menu, and you launch a custom process. And so on. Those buttons and menu items are navigation elements.

The design of the navigation system, the user interface in Lightning Experience, is very different from Salesforce Classic. We're not going to talk about those differences here, but you'll want to be familiar with where everything moves when you switch between the two user experiences. You can learn more about that right here in Trailhead, in the Navigating Lightning Experience and Setup unit.

Another, less visible kind of navigation is the "something happens" part of the above. Behind the scenes, Salesforce decides what's going to happen when you select an item in a menu, or click a link or button. Much of this navigation is built into Salesforce already, while other aspects are customizable—for example, overriding actions with Visualforce pages. But all of this navigation is managed by code written by Salesforce.

And then there's navigation in your own apps—apps that use *your* code to control application flow. When your custom action opens a form and the user clicks save, where do you go? When your running code makes a decision about where the user should go next, and sends them there. *This* is what we're going to talk about in this unit.

Classic Visualforce Navigation

"Classic" Visualforce navigation can be boiled down to "what happens at the end of an action method." Action methods return a `PageReference` object with the details of where the user is to be navigated to, and then the Visualforce framework handles the details of sending the right response back to the user's browser. And, great news, all of this still works.

Remember also that the Standard Controller returns a `PageReference` from its action methods. So, your existing navigation, whether you're using the Standard Controller or your own custom controller code, continues to work as you expect.

Modern Visualforce Navigation

So, if classic Visualforce navigation works, why are we still talking about this? What are we even having a conversation about? We just want to say one word to you. Just one word. Are you listening? . . . "JavaScript."

There's a great future in JavaScript—and that future is here today. Many Visualforce developers are using JavaScript heavily in their apps, and that use continues to grow. Classic Visualforce works, and will continue to work for a long time. But as developers adopt Visualforce features such as Remote Objects and JavaScript remoting, more of their apps' behavior migrates from the server side, where `PageReference` is the rule, to the browser and JavaScript, where there's no such thing as a `PageReference`.

In the Lightning Experience (and Salesforce1) world, there are rules and tools for building navigation in JavaScript. We'll cover the rules, which are mostly about what not to do, in a little bit. Let's talk about the right way to do things first.

Lightning Experience manages navigation using events. The navigation event framework is made available as a JavaScript utility object that provides a number of functions that make creating programmatic navigation straightforward. The `sforce.one` object is automatically added to Visualforce pages when they run in Lightning Experience. This object provides a number of functions that trigger navigation events when the functions are called. To use these functions, you can call them directly from your page's JavaScript code, or you can attach calls as click (or other) handlers to elements on the page.

Important: The `sforce.one` object isn't available in Salesforce Classic. Any code that uses it should test for the existence of `sforce.one` first.

The `sforce.one` object provides the following functions. Reference the function using dotted notation from the `sforce.one` object. For example: `sforce.one.navigateToSObject(...)`.

Function	Description
`back([refresh])`	Navigates to the previous state that's saved in the `sforce.one` history. It's equivalent to clicking a browser's Back button.
`navigateToSObject( recordId[, view])`	Navigates to an sObject record, specified by `recordId`.
`navigateToURL(url[, isredirect])`	Navigates to the specified URL.
`navigateToFeed( subjectId, type)`	Navigates to the feed of the specified `type`, scoped to the `subjectId`.
`navigateToFeedItemDetail( feedItemId)`	Navigates to the specific feed item, `feedItemId`, and any associated comments.

Function	Description
`navigateToRelatedList( relatedListId, parentRecordId)`	Navigates to a related list for the `parentRecordId`.
`navigateToList( listViewId, listViewName, scope)`	Navigates to the list view that's specified by the `listViewId`, which is the ID of the list view to be displayed.
`createRecord( entityName[, recordTypeId])`	Opens the page to create a new record for the specified `entityName`, for example, "Account" or "MyObject__c".
`editRecord(recordId)`	Opens the page to edit the record specified by `recordId`.

For additional details about using these functions, and the parameters they accept, see Navigation with the `sforce.one` Object in this unit's Resources.

Navigation Gotchas, and How to Fix Them

The first rule for building Visualforce navigation in JavaScript is: do not set `window.location` directly. The second rule for building Visualforce navigation in JavaScript is: *do not* set `window.location` directly.

Don't Set `window.location` Directly

OK, gratuitous repetition and movie reference aside, what's the big deal here? It's pretty simple. When in Lightning Experience your page doesn't *have* a `window.location` to set! Remember all the earlier discussion about the Visualforce "container," and being in an iframe, and Lightning Experience being some kind of health club? (SPA—single-page application.) This is one of the things that falls out of it. The Visualforce iframe doesn't have direct access to the `window.location` value, so you can't set it. If your code depends on setting it, it'll break. That is, actions that fire navigation by setting `window.location` will simply stop navigating to wherever you were expecting to go.

There's actually a way around this restriction, but you shouldn't use it. The reason is if you bypass the navigation functions in `sforce.one`, your navigation events won't be tracked in the Lightning Experience navigation stack. That stack provides useful features, like Back buttons that account for redirects and the like. A number of features in Lightning Experience (and especially in Salesforce1) depend on that stack containing all navigation events. It's worth making sure you use it correctly.

The Salesforce Classic Issue

So, yeah, there's this one…thing. Unfortunately, the `sforce.one` utility object isn't available when your page runs in Salesforce Classic. In that context, you *have* to use `window.location`. The good news is, in Salesforce Classic, `window.location` is available. The bad news is, this limitation means you'll have to add an ugly `if` block to your code. Consider wrapping your navigation functions in utility methods that deal with this complexity, so that your main navigation logic can be straightforward.

Static URLs

Don't use static URLs to Salesforce resources. That is, if you're adding a link to edit a Contact record, don't create the link by building a string with a static pattern like `link = '/' + accountId + '/e'`. In some contexts this works, but in others it doesn't. Instead, try one of these approaches:

- In Visualforce markup, use `{!URLFOR($Action.Contact.Edit, recordId)}`
- In JavaScript, use `navigateToSObject(recordId)`

There are actions and functions for viewing, creating, editing, and so on. Use them, rather than URL strings.

CHAPTER 34 Understanding Important Visual Design Considerations

In this chapter ...

- Affecting the Styling of Standard Components
- Styling Strategies and Recommendations
- The Salesforce Lightning Design System

Visualforce pages look the same whether they are running in Salesforce Classic or Lightning Experience, unless you rework them to adapt to the appropriate user interface context. Built-in Visualforce components that display user interface elements aren't easily restyled to match the Lightning Experience look-and-feel.

Specifically, the HTML that's rendered by the built-in Visualforce components doesn't change when the page displays in Lightning Experience, and the Salesforce Classic stylesheets those components use is, by default, loaded by the page. The effect is that pages that use `<apex:inputField>`, `<apex:outputField>`, the `<apex:pageBlock>` components, and other coarse- and fine-grained components that match the Salesforce Classic visual design, still match that visual design. You get a little slice of Salesforce Classic in the middle of your Lightning Experience.

It's our general recommendation that—for now, for existing pages—you don't try to adapt them to match the visual design of Lightning Experience. Lightning Experience is still evolving, and matching its styling yourself means you're chasing a moving target. That's work.

Nevertheless, in some cases you'll want some pages to match more closely with Lightning Experience visuals. For new pages, or if you're willing to do some work, there are some great tools for creating pages that fit in perfectly with Lightning Experience.

Affecting the Styling of Standard Components

Visualforce provides a range of options for adjusting or overriding the styling of the standard components. If your goal is to make modest changes to the appearance of these components, the effort to use these options is similarly modest. Let's look at a few of the tools you have available for affecting styling.

Styling Individual Components

Visualforce components that produce HTML have pass-through `style` and `styleClass` attributes. These attributes allow you to use your own styles and style classes to control the look and feel of the resulting HTML. `style` allows you to set styles directly on a component, while `styleClass` lets you attach classes for styles defined elsewhere. For example, the following code sets the class of the `<apex:outputText>` and applies a style.

```
<apex:page>

    <style type="text/css">
        .asideText { font-style: italic; }
    </style>

    <apex:outputText style="font-weight: bold;"
        value="This text is styled directly."/>

    <apex:outputText styleClass="asideText"
        value="This text is styled via a stylesheet class."/>

</apex:page>
```

Adding a Custom Stylesheet

You can add your own custom stylesheets to any Visualforce page using static resources and the `<apex:stylesheet>` tag. For example, to add a stylesheet that's been uploaded as a static resource named "AppStylesheet", add the following to your page.

```
<apex:stylesheet value="{!$Resource.AppStylesheet}"/>
```

You can then refer to any of the styles contained in the stylesheet, and reference them in Visualforce tag `styleClass` attributes, as we did with the `asideText` style previously.

This is the recommended method for adding CSS style definitions to Visualforce pages, because it shares the stylesheet between pages, and minimizes the markup you need to add to each page.

The exception to this method for adding a stylesheet is the Salesforce Lightning Design System. The Lightning Design System is a fantastic all-new toolkit for styling your pages, and we'll talk about it in detail shortly.

Although you can upload the Lightning Design System as a static resource and reference it using `<apex:stylesheet>`, there's an easier way: Just include `<apex:slds />` anywhere in the markup of your page.

Different Styles in Lightning Experience

To load a custom stylesheet only when your page is running in Lightning Experience, use the following markup. This is similar to the Visualforce markup example in Sharing Visualforce Pages Between Classic and Lightning Experience.

```
<apex:page standardController="Account">

    <!-- Base styles -->
    <apex:stylesheet value="{!URLFOR($Resource.AppStyles,
'app-styles.css')}" />

    <!-- Lightning Desktop extra styles -->
    <apex:variable var="uiTheme" value="lightningDesktop"
        rendered="{!$User.UIThemeDisplayed == 'Theme4d'}">
        <apex:stylesheet value="{!URLFOR($Resource.AppStyles,
'lightning-styling.css')}" />
    </apex:variable>

    <!-- Rest of your page -->

</apex:page>
```

OK, these are tools. Let's look at a few techniques for using them next.

Styling Strategies and Recommendations

To create Visualforce pages that match the Lightning Experience visual design, create new pages using the Lightning Design System. There are a couple of ways to use the Lightning Design System in your Visualforce pages.

Before we get to specifics, let's think at a higher level and consider the different strategies for applying Lightning Experience styling to your pages. In particular, let's talk about your existing pages.

There are two ways to affect the styling of existing pages to make them look more like Lightning Experience.

- Change the *markup* to apply new styling—make changes in your pages.
- Change the *styling rules* for existing markup—make changes in your stylesheets.

These aren't either / or. You can use them individually or in combination.

Correctly using the Lightning Design System means using the Lightning Design System stylesheets with all-new markup for your Visualforce pages. Again, this is the only *supported* method for matching the Lightning Experience visual design.

To do this, you can either download the Lightning Design System stylesheets from their website and use them as you would any other stylesheet, or you can add the `<apex:slds>` component to the markup of your Visualforce page. The `<apex:slds>` component allows you to reference Lightning Design System stylesheets without uploading them as a static resource, simplifying the syntax of your page and preventing you from hitting the 250-mb static resource limit.

Using `<apex:slds>` comes with its own set of guidelines and considerations. If you want to know more, work your way through the Lightning Design System badge or check out the link to the *Visualforce Developer Guide* in the Resources section.

It's also possible to add the Lightning Design System stylesheets, and revise your pages to use them. How much work this is depends on how closely you want to match Lightning Experience as well as the specific markup and components in your code. While it's possible to achieve decent results this way, however, it's not an approach we recommend. The Lightning Design System was designed to be applied to specific markup, and that's simply not what Visualforce emits. There's an "impedance mismatch" that, while not fatal, is definitely a serious rock in your shoe when you take this path.

Finally, there's the other approach: adding new rules and styles to your existing (or a new) stylesheet to make your existing markup look more like Lightning Experience. If your page is already mostly styled with your own stylesheets, this approach might work well for you. If instead you're mostly using the built-in Visualforce components and the Salesforce Classic styling, it requires you to override the styles from the Salesforce Classic stylesheet.

While this is technically possible, we want to discourage you from taking this approach. It introduces dependencies into your markup and styles that you don't want to have. These dependencies are on the structure, IDs, and classes of the HTML rendered by the built-in Visualforce components. We want to be really clear here: the HTML rendered by the built-in Visualforce components is an internal implementation detail, subject to change without notice. If you have dependencies on it in your own stylesheets, your styling *will* eventually break.

The Salesforce Lightning Design System

The Lightning Design System is a design framework for building enterprise apps that look like Lightning Experience. It includes a sophisticated CSS framework, a collection of graphic assets, and the Salesforce Sans font. You can use the Lightning Design System to build pages and apps that look gorgeous and perfectly match the Lightning Experience user interface.

The Lightning Design System was designed to make it easy for customers and partners to match the Lightning Experience look and feel. It also includes tools that make it possible to customize the look and feel to match your own brand—colors and so on—while still remaining consistent with overall Lightning Experience design.

The Lightning Design System is so big and so exciting that. . . we're not going to go into the details of using the Lightning Design System here. Because we've written a whole module about using it, Lightning Design System. It explains how to get the Lightning Design System, the basic concepts of using it to design pages, and how to apply those concepts to building Lightning Experience apps with Visualforce.

Lightning Design System is a big module—you'll have to work to earn that badge. While we do want to save the details for that module, we don't want to leave you totally hanging here. So, let's cover how you use the Lightning Design System with Visualforce in a general way.

The first thing to know is that the Lightning Design System assumes a new markup structure and styling classes. For this reason it's best used with new pages and apps. It's built around the capabilities of modern browsers, and takes advantage of the latest best practices for markup and styling. As much as we all love it, Visualforce has been around a while. Between the HTML it generates and static code in customer pages, most organizations will find it challenging to apply the Lightning Design System to existing pages.

The Lightning Design System module is focused on creating new pages and apps, and scoring that badge is the best way to learn about it. After finishing that module you'll have an understanding of both how to use Lightning Design System and how to plan development around it.

CHAPTER 35 Knowing Which Features to Avoid in Lightning Experience

In this chapter ...

- Lightning Experience Header and Navigation Menu Can't Be Suppressed
- Salesforce Classic Header and Sidebar are Always Suppressed
- <apex:relatedList> and Blacklisted Related Lists
- Avoid <apex:iframe>
- No, Really, Don't Set window.location Directly
- sforce.one Isn't Salesforce1-Only
- Changes With Action Overrides

There are a limited number of Visualforce components that we recommend you avoid on pages used in Lightning Experience. Additionally, a few features of Visualforce behave differently when used in Lightning Experience. Finally, there are a few places in Lightning Experience where you can't use Visualforce pages or apps, or where they might not function as expected.

Lightning Experience is still evolving and growing, and—Safe Harbor alert!—we hope to shrink this list as time goes on.

Lightning Experience Header and Navigation Menu Can't Be Suppressed

Visualforce pages always display with the standard Lightning Experience user interface when they run in Lightning Experience. There's no way to suppress or alter the Lightning Experience header or sidebar. In particular, the `showHeader` and `showSidebar` attributes of `<apex:page>` have no effect on Visualforce pages when displayed in Lightning Experience.

This behavior is intentional. Apps that display in Lightning Experience are *Lightning Experience* apps. If you need to provide a completely custom interface for your app, you'll need to run it in Salesforce Classic.

Salesforce Classic Header and Sidebar are Always Suppressed

The standard Salesforce Classic header and sidebar are always suppressed for pages when they're displayed in Lightning Experience. In particular, the `showHeader` and `showSidebar` attributes of `<apex:page>` have no effect on Visualforce pages when displayed in Lightning Experience.

Pages behave as though the `showHeader` and `showSidebar` attributes of `<apex:page>` are both set to `false`.

Note: The `standardStylesheets` attribute of `<apex:page>`, which determines whether to include or suppress the standard Salesforce Classic stylesheets, is unaffected by Lightning Experience. That is, it defaults to `true` in Lightning Experience, but you're able to change it.

`<apex:relatedList>` and Blacklisted Related Lists

There are a number of related lists that aren't supported in Lightning Experience. These related lists are "blacklisted," which means they are explicitly prevented from being used. As you might expect, these same related lists are blacklisted in Visualforce with the `<apex:relatedList>` tag.

See "Data Access and Views: What's Different or Not Available in Lightning Experience" in the online help for details of which related lists aren't supported in Lightning Experience.

Avoid `<apex:iframe>`

While it's not impossible to use `<apex:iframe>` on a Visualforce page in Lightning Experience, we recommend avoiding it.

Visualforce pages are wrapped in their own iframe when displayed in Lightning Experience. As discussed at length in Exploring the Visualforce App Container, this has a number of significant implications for how the page behaves. Adding an additional level to the iframe stack increases the complexity of the environment.

If you really understand iframes and how they affect the DOM and JavaScript, you can manage this complexity. But unless you've already been working with nested iframes, it's more likely that you'll have difficult to debug problems. For this reason, we suggest you avoid this tag on pages that are used in Lightning Experience.

No, Really, Don't Set `window.location` Directly

We probably sound like a broken record at this point, but it's important. If your page's JavaScript code is setting the `window.location` variable directly, that won't work when the page is displayed in Lightning Experience. You *must* modify this code for the page to work in Lightning Experience.

See the Managing Navigation unit for details.

`sforce.one` Isn't Salesforce1-Only

The `sforce.one` JavaScript utility object is available to Visualforce pages in both Salesforce1 and Lightning Experience. If you've been using the presence of the `sforce.one` object as a way to tell if your page is running in a mobile or desktop context, you need to update your code.

Use one of the documented methods to distinguish between the Salesforce Classic, Salesforce1, and Lightning Experience environments. Supported techniques are available in Visualforce, Apex, and JavaScript.

For the full details, see the Sharing Visualforce Pages Between Classic and Lightning Experience unit.

Changes With Action Overrides

Perhaps the most significant change, in terms of things that might be hard to work around, Visualforce overrides of standard actions are slightly different in Lightning Experience compared to Salesforce Classic. Any override for the object list action won't be accessible in Lightning Experience.

Specifically, there are six standard actions you can override for an object in Salesforce Classic:

- Object tab
- Object list
- Record view
- Record edit
- Record create
- Record delete

In Lightning Experience the first two actions have been combined into one page, object home. Object home is similar to object list, with some elements of object tab, like recent items, added. Others, such as reports or tools, have moved to other parts of the user interface.

Regardless of the user interface settings in your organization, both object tab and object list are available to be overridden in Setup. Overriding the object tab action overrides the object home page in Lightning Experience, as expected.

However, when in Lightning Experience, the object list action isn't accessible in the user interface, so there's no way to fire it. If your organization has overridden the object list action for any object, that functionality won't be available when users are using Lightning Experience. If there are essential features in that override, you'll need to find another way to make them available.

This table lists the standard actions you can override for an object in Setup, and the action that's overridden in the three different user experiences.

Override in Setup	Salesforce Classic	Lightning Experience	Salesforce1
Tab	object tab	object home	search
List	object list	**n/a**	object home
View	record view	record home	record home
Edit	record edit	record edit	record edit
New	record create	record create	**n/a**
Delete	record delete	record delete	record delete

Note: "n/a" doesn't mean you can't *access* the standard behavior, and it doesn't mean you can't *override* the standard behavior. It means you can't *access the override*. It's the override's functionality that's not available.

INDEX

A

Account Insights 34, 68
Accounts 8, 72
Actions
- in Lightning Experience 102, 107
- in the Global Actions menu 102
- on list view items 102
- on record pages 102
- on related lists 102
- on the Chatter page 102
- on the Home page 102

Activity Timeline 60
Apex 29
App Launcher
- overview 111

App Manager
- overview 113

App Menu 43
AppExchange apps 28
Apps
- lightning 109–111, 113
- overview 109–111, 113

Assessing the impact of Lightning Experience 30
Assistant 68

C

Calendar 77
Case feed 72
Cases 72
Chatter 81
Classic apps
- upgrade to Lightning apps 121–122, 124

Compact layouts 95, 98
Comparing the two interfaces 3, 5, 15, 19, 29
Considerations 33
Contacts 8, 72
Creating Lightning Experience apps 111, 117–118
Custom apps
- lightning 109–111, 113, 117–118

Custom buttons
- JavaScript 155–156, 159, 161, 166–168, 171, 173, 181
- migrate functionality to Lightning Experience 155–156, 159, 161, 166–168, 171, 173, 181
- support in Lightning Experience 155–156

Custom help links 56
Custom objects 72
Custom URL buttons 166–167
Custom Visualforce buttons 168
Customization
- programmatic 29

Customizing 43, 98
Customizing Lightning Experience 95

D

Duplicate management 34

E

Enabling by edition 35
Enabling Lightning Experience 33–34, 37
Enabling users 35
Enhanced Notes 34
Events 77

F

Feed 81

G

Get your organization ready 34
Global search 52

Global variables
- $User.UITheme 268
- $User.UIThemeDisplayed 268

H

Help menu 56
Highlights panel 98
Home 8, 68

I

Is Lightning Experience right for me? 15, 29

J

JavaScript custom buttons
- security issues 156

L

Lead conversion 34
Lead Workspace 65
Lightning actions
- 3rd-Party API Integration 181
- populate fields based on user input 173
- provide feedback messages to users 173

Lightning Apps
- create 117–118
- navigation bar 110
- overview 109–111, 113
- upgrade from Classic app 121–122, 124

Lightning component actions 171, 173, 181
Lightning Experience
- JavaScript custom button support 155–156
- standard objects 119

Lightning Experience Developer Edition 191
Lightning Experience features chart 188
Lightning Experience overview 1–2, 7
Lightning Experience Rollout Strategy 33
Lightning Experience Supported Editions 18
Lightning Experience Supported User Licenses 18
Lightning Setup page 37
Limitations 50, 70
List views 63, 70
List Views 8

N

navigation
- create menu 119

Navigation menu 41

O

Object home 60, 65, 70
Object Manager 50
Opportunity home 63
Opportunity Kanban 8, 63
Opportunity Workspace 8, 60
Other enhancements 67

P

Page layouts
- how they work in Lightning Experience 96

Performance chart 68
Permission sets 35
Piloting Lightning Experience to your users 35
Preview Lightning Experience 30

Q

Quick actions 161

R

Recommended features 34
Reports and Dashboards
- dashboard components 89
- dashboard editor 88
- feature comparison 93
- interactive charts and filters 90
- interactive dashboards 89
- new home pages 88
- overview 87
- report run page 90

Rolling out Lightning Experience
- communication plan 205
- crafting a communication strategy 205
- crafting a marketing strategy 205
- create a Chatter group 201
- creating a project schedule 203
- creating a training plan 206
- decide who to roll out to 198
- define measures for success 204
- deliver an executive summary 214
- demo 192–193
- educate your company 210
- educating your company 191–194, 196, 199
- Educating your company 192
- educating yourself 188, 191, 194
- employee survey 211
- enabling Lightning Experience 198, 209
- engage with super users 201, 210
- engaging your super users 199
- executive sponsor 191, 199, 201, 214
- gap analysis 194, 196
- go live 188, 196–198, 202, 205, 207, 209–211, 214–215
- identifying stakeholders 191
- Lightning Setup page 209
- measure results 204
- measuring results 211, 213
- overview 187
- picking a launch date 202
- planning your rollout 198
- presenting your findings 196
- ready to start 207
- revisiting your processes 193
- risk severity matrix 196
- rollout checklist 198
- rollout strategy 188, 196–199, 201–203, 205–207, 209–211, 213–214
- super users 199, 210
- test your customizations 207
- useful Trailhead modules 188

Rolling out Lightning Experience *(continued)*
- where to start 198

Rollout overview 198

S

Sales in Lightning Experience 59
Sales path 34
Sales Path 60, 65
Salesforce Classic 3, 5, 17
Search 52
Setup 41, 43, 50, 52, 56
Shared Activities 34
Social Accounts, Contacts, and Leads 34
Supported vs. unsupported features 19, 29
Switcher 39
Switching between Lightning Experience and Salesforce Classic 39

T

Tasks 77
Turn it on! 37

U

Understanding both user interfaces 17
Unsupported features 28
Upgrade Classic Apps
- overview 121
- task 122
- test 124

User profile 76

V

Visualforce 29, 227–229, 231, 245, 247, 253–254, 258–259, 261–263, 265, 267–268, 270, 274–275, 277–280, 283–285, 287, 289–291

W

What is Lightning Experience? 2
What’s new 59