



---

# The Lightning Experience Guide

How to Transition to the New Salesforce

Version 8, Winter '19

**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**

© Copyright 2000–2018 salesforce.com, inc. All rights reserved. Salesforce is a registered trademark of salesforce.com, inc., as are other names and marks. Other marks appearing herein may be trademarks of their respective owners.

Various trademarks held by their respective owners.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior consent of the publisher.

# CONTENTS

<b>WELCOME TO LIGHTNING EXPERIENCE</b> .....	<b>1</b>
<b>Chapter 1: What Your Sales and Service Teams Need</b> .....	<b>1</b>
Welcome to Lightning Experience .....	2
What About the Previous User Experience? .....	3
<b>Chapter 2: What Makes Lightning Experience So Special</b> .....	<b>7</b>
Take a Tour of Lightning Experience .....	8
<b>Chapter 3: How Different Personas Benefit from Lightning Experience</b> .....	<b>17</b>
<b>Chapter 4: Understanding Lightning Experience and Salesforce Classic</b> .....	<b>19</b>
Does My Org Qualify for Lightning Experience? .....	20
Comparing Lightning Experience and Salesforce Classic .....	21
Other Features That Aren't Supported .....	37
What About Apex and Visualforce? .....	38
Evaluate Your Org with the Lightning Experience Readiness Check .....	38
Preview Your Org in Lightning Experience .....	39
<b>Chapter 5: Your Lightning Experience Rollout Strategy</b> .....	<b>41</b>
Enabling Lightning Experience .....	42
Recommended Features for Lightning Experience .....	42
Set Up Users for Lightning Experience .....	43
Turn Lightning Experience On! .....	45
Switching Between Lightning Experience and Salesforce Classic .....	47
<b>WHAT'S NEW IN LIGHTNING EXPERIENCE</b> .....	<b>51</b>
<b>Chapter 6: Navigation and Setup in Lightning Experience</b> .....	<b>51</b>
The Lightning Experience Navigation Bar .....	52
The App Launcher in Lightning Experience .....	55
Search for Records .....	58
Lightning Experience Help Menu .....	64
Meet the New and Improved Setup .....	68

## Contents

Object Manager	72
Service Setup Flows	74
<b>Chapter 7: Opportunities, Leads, and Selling in Lightning Experience</b>	<b>75</b>
Explore the Opportunity Workspace	76
Opportunity Home	79
Explore the Lead Workspace	80
Accounts and Contacts in Lightning Experience	81
<b>Chapter 8: Service Cloud in Lightning Experience</b>	<b>87</b>
The Fast-As-Lightning Service Console	88
Be a Knowledge Sponge	89
Snap-ins Chat for Fast, Snappy Service	90
Analyze the Service Analytics App	93
Get Out There with Field Service Lightning	94
Omni-Channel Knows All	95
Social Customer Service	96
<b>Chapter 9: Explore More of Lightning Experience</b>	<b>99</b>
Custom Themes and Branding	100
The Home Page	100
Object Home Pages and List Views	103
How Other Objects are Organized in Lightning Experience	105
Chatter in Lightning Experience	108
File Management and Collaboration	111
Tasks, Calendar, and Events	116
<b>Chapter 10: Reports and Dashboards in Lightning Experience</b>	<b>121</b>
Report and Dashboard Home Pages	122
Create Beautiful Dashboards Using a Feature-Rich Editor	123
Present and Share Information in Interactive Dashboards	124
Get More Information from Interactive Charts and Filters on the Report Run Page	125
Reports and Dashboards: Compatibility Between Lightning Experience and Salesforce Classic	128



<b>HOW YOUR EXISTING CUSTOMIZATIONS AFFECT LIGHTNING EXPERIENCE</b> .....	<b>131</b>
<b>Chapter 11: Your Layouts Can Customize Lightning Experience Records</b> .....	<b>131</b>
Page Layouts in Lightning Experience .....	<b>132</b>
Compact Layouts in Lightning Experience .....	<b>134</b>
<b>Chapter 12: Actions and Lightning Experience</b> .....	<b>137</b>
Actions in Lightning Experience .....	<b>138</b>
How Actions Are Ordered in Lightning Experience .....	<b>144</b>
<b>CREATE LIGHTNING APPS</b> .....	<b>145</b>
<b>Chapter 13: What Is a Lightning App?</b> .....	<b>145</b>
Navigate at the Speed of Lightning .....	<b>146</b>
Find Items and Apps in the App Launcher .....	<b>149</b>
Meet the Lightning Experience App Manager .....	<b>152</b>
Install a Custom Classic App: The Warehouse Data Model .....	<b>153</b>
<b>Chapter 14: Step into the World of Lightning Apps</b> .....	<b>155</b>
Create a Lightning App .....	<b>156</b>
Create Apps for Different Audiences .....	<b>157</b>
<b>Chapter 15: Take Your Classic App to the Next Level: Lightning</b> .....	<b>161</b>
Upgrade the Warehouse App .....	<b>162</b>
Give the Warehouse App That Touch of Lightning .....	<b>162</b>
Look at the Warehouse App Now .....	<b>164</b>
<b>CREATE CUSTOM PAGES FOR LIGHTNING EXPERIENCE</b> .....	<b>167</b>
<b>Chapter 16: Meet the Lightning App Builder</b> .....	<b>167</b>
How the Lightning App Builder Works .....	<b>168</b>
The Lightning App Builder User Interface .....	<b>169</b>
Lightning Page Types .....	<b>171</b>
<b>Chapter 17: Yes, Virginia, You Can Customize Your Lightning Experience Home Page</b> .....	<b>173</b>

## Contents

Create a Custom Home Page . . . . .	175
Roll Out Your Custom Home Page to Your Lightning Experience Users . . . . .	176
Look at Your Handiwork . . . . .	176

### **Chapter 18: Those Lightning Experience Record Pages? You Can Customize**

<b>Them, Too!</b> . . . . .	179
Create a Custom Lightning Record Page . . . . .	180
Make Your Record Page Dynamic . . . . .	183
Roll Out Your Custom Record Page to Your Lightning Experience Users . . . . .	184
See What You Did There? . . . . .	186

### **Chapter 19: The App Home Lightning Page** . . . . . 189

Create an App Page . . . . .	190
Add Your App Page to Lightning Experience and the Salesforce Mobile App . . . . .	191
Test Your App Page in the Salesforce Mobile App . . . . .	193
Test Your App Page in Lightning Experience . . . . .	194

### **Chapter 20: Custom Lightning Components in the Lightning App Builder** . . . . . 197

My Domain Is Already On in Your Trailhead Playground . . . . .	199
Install a Custom Lightning Component . . . . .	199
Add the Custom Lightning Component to Your App Page . . . . .	200
Test the Custom Component . . . . .	200

## **MIGRATE JAVASCRIPT BUTTON FUNCTIONALITY TO LIGHTNING** . . . . . 203

### **Chapter 21: JavaScript Buttons: It's Time to Move On** . . . . . 203

JavaScript Button Security Issues and Use Cases . . . . .	204
---	-----

### **Chapter 22: You Have Buttons? We Have Alternatives** . . . . . 207

Quick Actions . . . . .	209
Custom URL Buttons and Links . . . . .	214
Apex Triggers . . . . .	215
Custom Visualforce Buttons . . . . .	216

### **Chapter 23: Lightning Actions: Smart, Fast, and Mobile** . . . . . 219

Populate Fields Based on User Input and Provide Feedback Messages to Users . . . . .	221
Integrate Third-Party APIs . . . . .	228
<b>ROLL OUT LIGHTNING EXPERIENCE . . . . .</b>	<b>235</b>
<b>Chapter 24: Welcome to Your Lightning Experience Rollout . . . . .</b>	<b>235</b>
What It Takes to Go Live . . . . .	236
Educate Yourself About Lightning Experience . . . . .	236
Identify Stakeholders and an Executive Sponsor . . . . .	239
Revisit Your Processes . . . . .	240
Perform a Gap Analysis . . . . .	241
Improve Your Implementation with Salesforce Optimizer . . . . .	243
Get Hands-On with Lightning Experience . . . . .	244
Assess the Impact . . . . .	244
Educate Your Company About Lightning Experience . . . . .	245
How to Demonstrate Lightning Experience . . . . .	246
When the Show Is a Go . . . . .	246
<b>Chapter 25: All Systems Go . . . . .</b>	<b>247</b>
Where to Start . . . . .	248
Plan Your Rollout . . . . .	248
Decide How to Structure Your Rollout . . . . .	248
Unleash the Power of Super Users . . . . .	252
Create a Chatter Group for the Rollout Team . . . . .	254
Pick a Launch Date . . . . .	255
Create a Project Schedule . . . . .	256
Define Measures for Success . . . . .	257
Create a Marketing and Communication Strategy . . . . .	258
Create a Training Plan . . . . .	259
Test Your Customizations and Iterate . . . . .	260
Ready to Start . . . . .	260
<b>Chapter 26: Go Live . . . . .</b>	<b>261</b>
Drive Adoption by Optimizing Your Implementation . . . . .	262
Engage with Super Users . . . . .	262
Measure Results . . . . .	264

## Contents

Survey Your Users . . . . .	264
Use Reports and Dashboards in Salesforce . . . . .	266
Deliver an Executive Summary . . . . .	267
The Show Must Go On . . . . .	268
<b>LIGHTNING EXPERIENCE DEVELOPMENT . . . . .</b>	<b>269</b>
<b>Chapter 27: User Interface Development Considerations . . . . .</b>	<b>269</b>
Raising the Bar for Web App User Interfaces . . . . .	270
Classic Visualforce . . . . .	270
Visualforce as a JavaScript Application Container . . . . .	271
Lightning Components . . . . .	272
Choosing the Right Tool for the Job . . . . .	273
Choosing the Right Tool for Your Organization . . . . .	276
Migrating to Lightning Components . . . . .	277
<b>Chapter 28: Visualforce and Lightning Experience . . . . .</b>	<b>279</b>
What Works . . . . .	280
What Works, but Needs Updating and Testing . . . . .	281
What Doesn't Work . . . . .	281
That Look-and-Feel Thing . . . . .	283
<b>Chapter 29: Lightning Components in Lightning Experience . . . . .</b>	<b>285</b>
Considerations for Use . . . . .	287
<b>Chapter 30: Salesforce Partners, Packaging, and AppExchange . . . . .</b>	<b>289</b>
Partner Tools in Lightning Experience . . . . .	290
Packaging Apps in Lightning Experience . . . . .	290
AppExchange and Lightning Experience . . . . .	291
<b>Chapter 31: Understand Changes to Other Development Tools . . . . .</b>	<b>293</b>
Installed Packages in Lightning Experience . . . . .	294
The API and Apex in Lightning Experience . . . . .	294
Authentication and Security in Lightning Experience . . . . .	294
Canvas for Lightning Experience . . . . .	294
Salesforce Mobile for Lightning Experience . . . . .	295

Mobile SDK for Lightning Experience . . . . .	295
<b>VISUALFORCE &amp; LIGHTNING EXPERIENCE</b> . . . . .	<b>297</b>
<b>Chapter 32: Use Visualforce in Lightning Experience</b> . . . . .	<b>297</b>
Where You Can Use Visualforce in Lightning Experience . . . . .	299
<b>Chapter 33: Developing Visualforce Pages for Lightning Experience</b> . . . . .	<b>305</b>
Set Up Your Editor . . . . .	306
Viewing Visualforce Pages During Development . . . . .	306
Reviewing Visualforce Pages in Multiple Environments . . . . .	310
Testing Your Visualforce Pages . . . . .	311
<b>Chapter 34: Exploring the Visualforce App Container</b> . . . . .	<b>313</b>
The Outer Lightning Experience Container . . . . .	314
The Visualforce iframe . . . . .	314
Impact of the New Container . . . . .	315
Visualforce Defaults and Environment Changes in Lightning Experience . . . . .	317
<apex:page> showHeader and sidebar Attributes Are Always false . . . . .	317
The sforce.one JavaScript Utility Object . . . . .	317
<b>Chapter 35: Sharing Visualforce Pages Between Classic and Lightning Experience</b> . . . . .	<b>319</b>
Detecting and Responding to the User Experience Context in Visualforce Markup . . . . .	320
Detecting and Responding to the User Experience Context in JavaScript . . . . .	322
Determining the User Experience Context in Apex . . . . .	323
Querying for Lightning Experience via SOQL and API Access . . . . .	324
<b>Chapter 36: Managing Navigation</b> . . . . .	<b>327</b>
Navigation in Lightning Experience . . . . .	328
Classic Visualforce Navigation . . . . .	328
Modern Visualforce Navigation . . . . .	329
Navigation Gotchas, and How to Fix Them . . . . .	330
<b>Chapter 37: Understanding Important Visual Design Considerations</b> . . . . .	<b>333</b>
Affecting the Styling of Standard Components . . . . .	334

## Contents

Styling Strategies and Recommendations .....	335
The Salesforce Lightning Design System .....	337
<b>Chapter 38: Knowing Which Features to Avoid in Lightning Experience</b> .....	<b>339</b>
Lightning Experience Header and Navigation Menu Can't Be Suppressed .....	340
Salesforce Classic Header and Sidebar Are Always Suppressed .....	340
<apex:relatedList> and Blacklisted Related Lists .....	340
Avoid <apex:iframe> .....	341
No, Really, Don't Set window.location Directly .....	341
sforce.one Isn't Salesforce Mobile-Only .....	341
Changes With Action Overrides .....	342
<b>INDEX</b> .....	<b>345</b>

# WELCOME TO LIGHTNING EXPERIENCE

## CHAPTER 1 What Your Sales and Service Teams Need

In this chapter ...

- [Welcome to Lightning Experience](#)
- [What About the Previous User Experience?](#)

Your sales and service 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 and deliver service 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.

Your service reps (or maybe you call them support agents) need these things so they can spend their time where it counts: helping your company's customers and answering questions quickly.

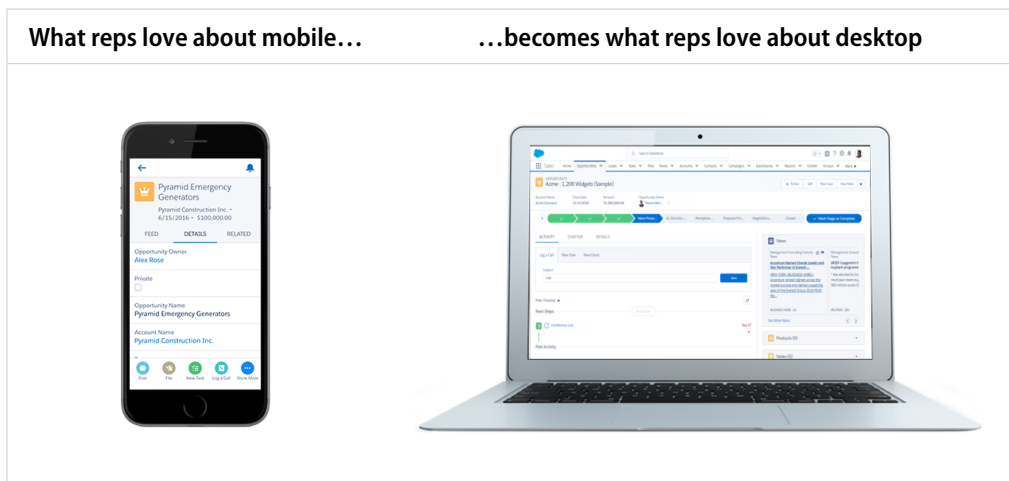
So how do you help your sales and service teams get what they need?

# Welcome to Lightning Experience

Welcome to Lightning Experience, the modern, beautiful user experience from Salesforce. With a sales- and service-centric mindset, we focused on reinventing the desktop environment to better support your business processes. The result is an intuitive, intelligent interface that helps sales and service teams work more naturally and productively.

Although we started with sales, Lightning Experience doesn't end there. Besides transforming Salesforce CRM, Lightning Experience also extends 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 Salesforce mobile experience and brings it to the desktop.



When we're talking about Lightning Experience, we're talking about pages in Salesforce that are optimized for sales and service use. We're talking about new features that help your sales and service teams focus on the right deals, the right customers, and the right activities, every time users log in. We're talking about flexible, interactive tools that help everyone visualize data and take care of business.

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 and service reps close deals and support customers 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 and service user. It's a simpler user experience, designed to help your sales team sell faster, with personalized alerts and an interactive assistant to keep sales reps focused on what's important. It's designed to help service reps support customers faster, bringing cases, knowledge, and support tools together in an efficient, problem-solving environment. All of this drives improved productivity for everyone in your org.

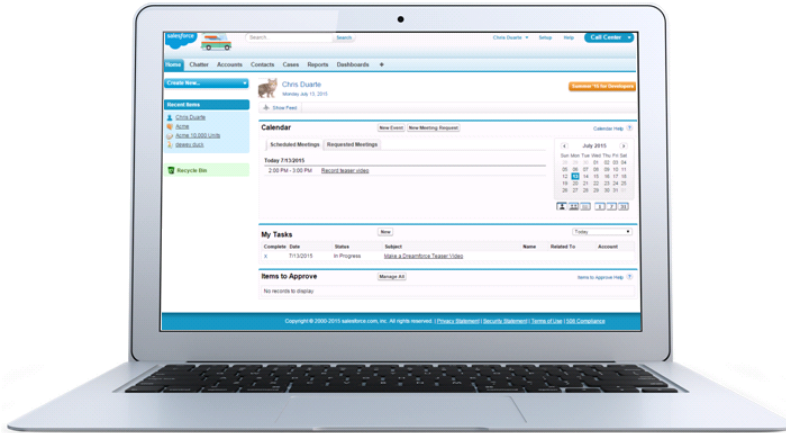
Ultimately, we built Lightning Experience because of you, our customer. Lightning Experience is the result of everything we've learned from you over the past years and releases. Lightning Experience is the killer sales and service app you can deliver to your teams.

## What About the Previous User Experience?

---

It's still here, and for the purposes of clarity, we're calling it Salesforce Classic.

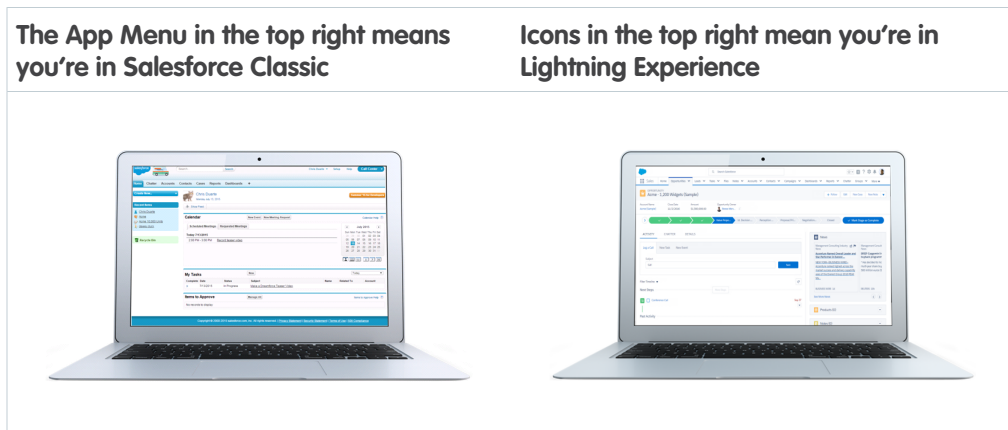
## Chapter 1 What Your Sales and Service Teams Need



When we say Salesforce Classic, we're talking about the venerable Salesforce interface that immediately predates Lightning Experience. It's the one with the App Menu and the user name, Setup, and Help links in the top right corner of the header. If you haven't customized your logo, you also see the seasonal Salesforce logo in the upper left corner.

We understand that you might not be ready to move off Salesforce Classic just yet. Depending on your company culture and size, change can take time. To help you manage the transition, we've made it easy for you to fire up Lightning Experience and keep Salesforce Classic around too. So you can ease your users over to the new way of doing things. We'll talk more about developing a phased rollout plan and change management in another module, since that's an important topic all on its own.

But now that you may be dealing with two Salesforce interfaces, you need to know how to tell when you're in one or the other.



It's important that you get comfortable with telling the two experiences apart so you're ready to help users with questions. After you enable Lightning Experience, one of the first things you'll need to identify on a troubleshooting call with a user is, are they in Lightning Experience or in Salesforce Classic?

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](#)

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

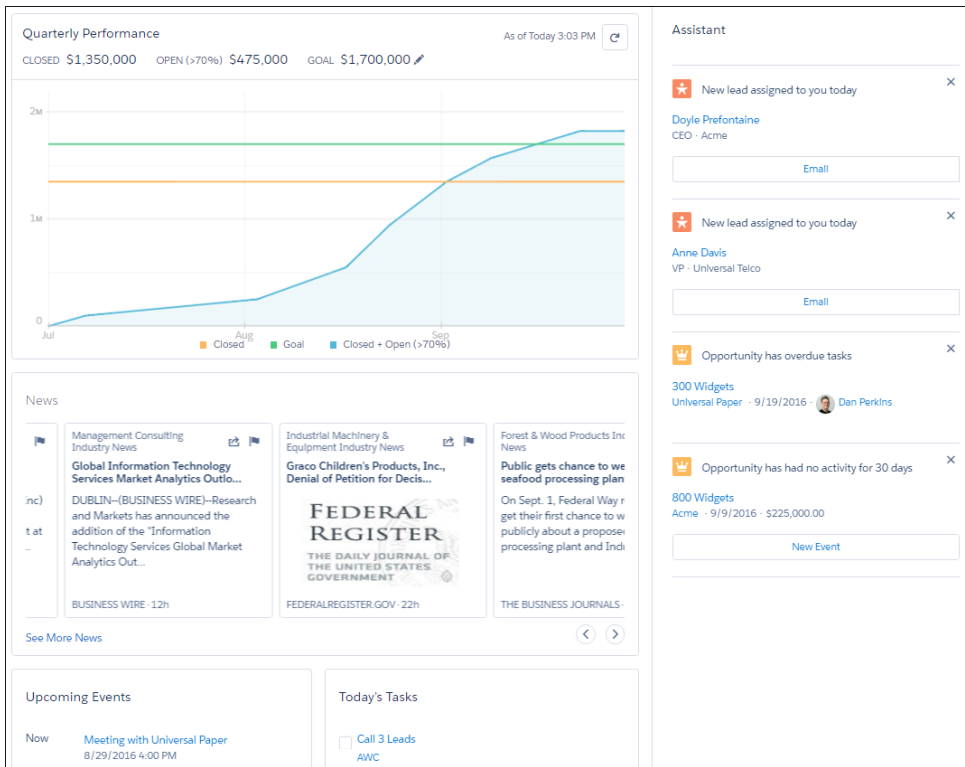
---

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.



## 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 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.

## Chapter 2 What Makes Lightning Experience So Special

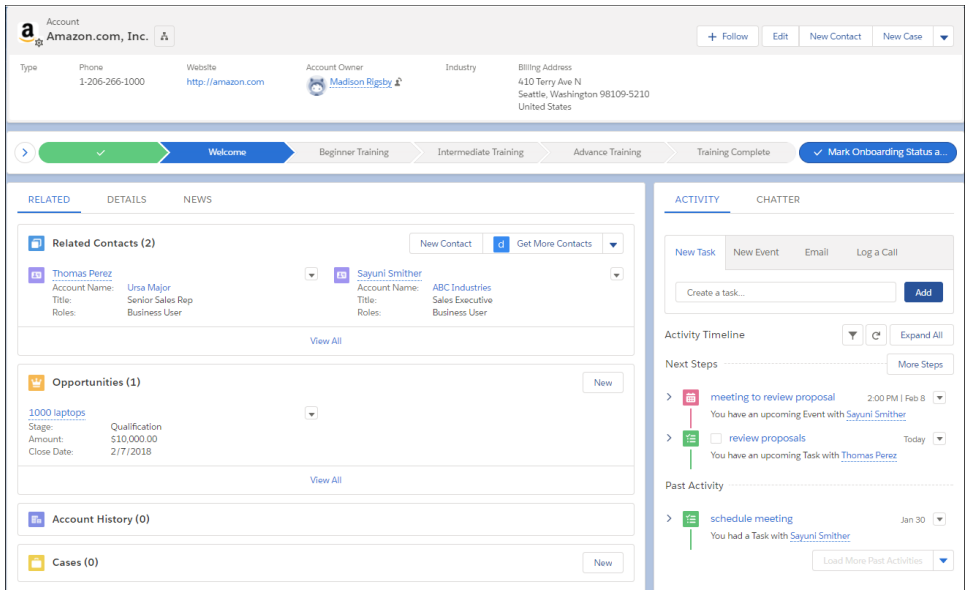
The screenshot displays a Salesforce Lightning Experience record for an Opportunity. At the top, the record title is "Acme - 1,200 Widgets (Sample)" with a "Follow" button and action buttons for "Edit", "Delete", and "Clone". Below the title, key fields are listed: ACCOUNT NAME (Acme (Sample)), CLOSE DATE (8/9/2015), AMOUNT (\$140,000.00), and OPPORTUNITY OWNER (Admin User). A progress bar shows the current stage as "Needs Analysis" (highlighted in blue), with previous stages "Discovery" and "Proposal" and future stages "Negotiation" and "Closed". A "Mark Stage as Complete" button is visible at the end of the bar. The "KEY FIELDS" section includes "Amount" (\$140,000.00), "Close Date" (8/9/2015), and a "Discovery Completed" checkbox. A "GUIDANCE FOR SUCCESS" section provides advice: "Understand the business need and decision criteria." with bullet points: "Why is our solution a good fit?", "How is our solution better than our competitors?", and "What resources are available to implement the solution?". Below this, the "ACTIVITY" tab is active, showing a "New Task" form with a "Subject" field and a "Save" button. A "Next Steps" section shows an event: "Internal opportunity team prep call" on Aug 18, 12:00 PM. On the right, a "Contact Roles (3)" sidebar lists: Howard Jo... (Decision Maker, Buyer), Edward Sta... (Economic Buyer, President and CEO), and Leanne To... (Influencer, 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 and service reps can find information and gather information at a glance.

- Locate important data efficiently with the redesigned Lightning Experience page layout.
- Get key coaching details on account records with a customizable Path to support your sales process.
- Get the latest news for your customers with News.
- Work smarter and keep your data clean with field-level duplicate matching.
- Review past and upcoming activities at a glance.



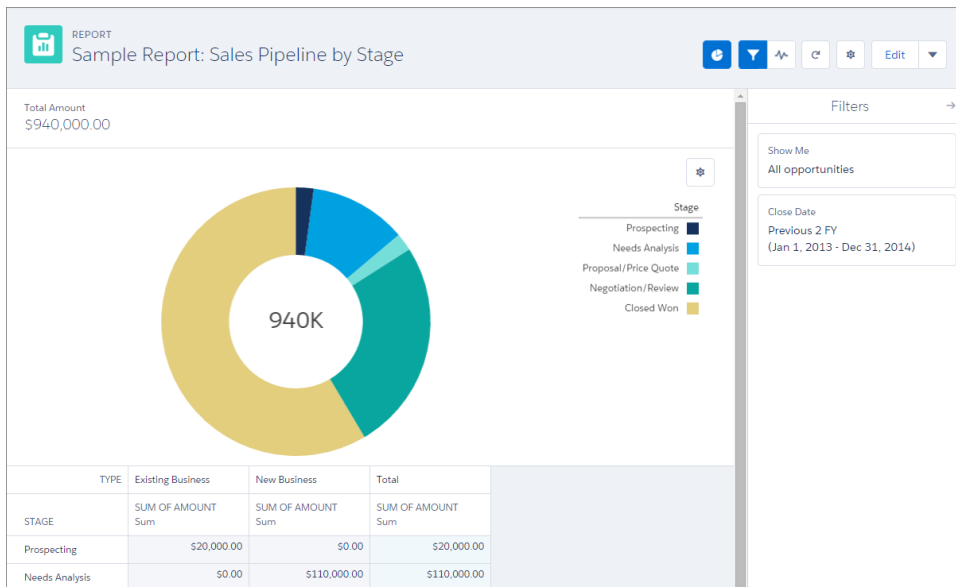


## Reports and Dashboards

Your users 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.
- Transition easily from Salesforce Classic to Lightning Experience with reports and dashboards that are automatically viewable in the new interface. And they inherit all permissions and sharing settings that were defined in Salesforce Classic.

## Chapter 2 What Makes Lightning Experience So Special



## List Views

Now your sales and service 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.
- Find the data you need in a snap with the list view search bar.
- Share your list views with user groups in your org.

Opportunities  
All Opportunities ▾

31 Items · Sorted by Opportunity Name · Filtered by all opportunities · Updated a minute ago

Search this list...

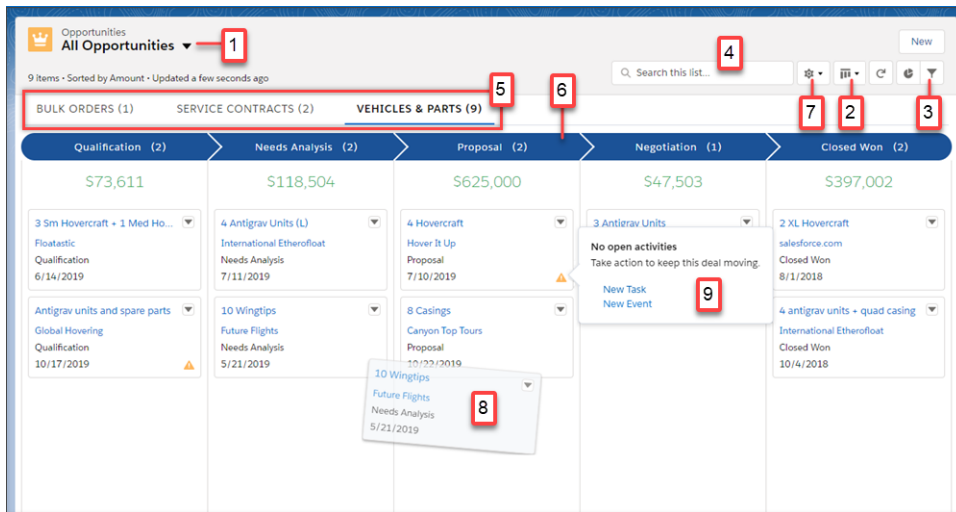
	OPPORTUNITY NAME ↑	ACCOUNT NAME	AMOUNT	CLOSE DATE	STAGE	OPPORTU...
1	<input type="checkbox"/> Burlington Textiles Weaving Plant Generator	Burlington Textiles Corp of America	\$235,000.00	3/21/2016	Closed Won	AUser ▾
2	<input type="checkbox"/> Dickenson Mobile Generators	Dickenson pic	\$15,000.00	3/21/2016	Qualification	AUser ▾
3	<input type="checkbox"/> Edge Emergency Generator	Edge Communications	\$75,000.00	3/21/2016	Closed Won	AUser ▾
4	<input type="checkbox"/> Edge Emergency Generator	Edge Communications	\$35,000.00	3/21/2016	Id. Decision Makers	AUser ▾
5	<input type="checkbox"/> Edge Installation	Edge Communications	\$50,000.00	3/21/2016	Closed Won	AUser ▾
6	<input type="checkbox"/> Edge SLA	Edge Communications	\$60,000.00	3/21/2016	Closed Won	AUser ▾
7	<input type="checkbox"/> Express Logistics Portable Truck Generators	Express Logistics and Transport	\$80,000.00	3/21/2016	Value Proposition	AUser ▾
8	<input type="checkbox"/> Express Logistics SLA	Express Logistics and Transport	\$120,000.00	3/21/2016	Perception Analysis	AUser ▾
9	<input type="checkbox"/> Express Logistics Standby Generator	Express Logistics and Transport	\$220,000.00	3/21/2016	Closed Won	AUser ▾
10	<input type="checkbox"/> GenePoint Lab Generators	GenePoint	\$60,000.00	3/21/2016	Id. Decision Makers	AUser ▾
11	<input type="checkbox"/> GenePoint SLA	GenePoint	\$30,000.00	3/21/2016	Closed Won	AUser ▾
12	<input type="checkbox"/> GenePoint Standby Generator	GenePoint	\$85,000.00	3/21/2016	Closed Won	AUser ▾
13	<input type="checkbox"/> Grand Hotels Emergency Generators	Grand Hotels & Resorts Ltd	\$210,000.00	3/21/2016	Closed Won	AUser ▾
14	<input type="checkbox"/> Grand Hotels Generator Installations	Grand Hotels & Resorts Ltd	\$350,000.00	3/21/2016	Closed Won	AUser ▾
15	<input type="checkbox"/> Grand Hotels Guest Portable Generators	Grand Hotels & Resorts Ltd	\$250,000.00	3/21/2016	Value Proposition	AUser ▾
16	<input type="checkbox"/> Grand Hotels Kitchen Generator	Grand Hotels & Resorts Ltd	\$15,000.00	3/21/2016	Id. Decision Makers	AUser ▾
17	<input type="checkbox"/> Grand Hotels SLA	Grand Hotels & Resorts Ltd	\$90,000.00	3/21/2016	Closed Won	AUser ▾

## Kanban View

The Kanban view organizes a set of records into columns to 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

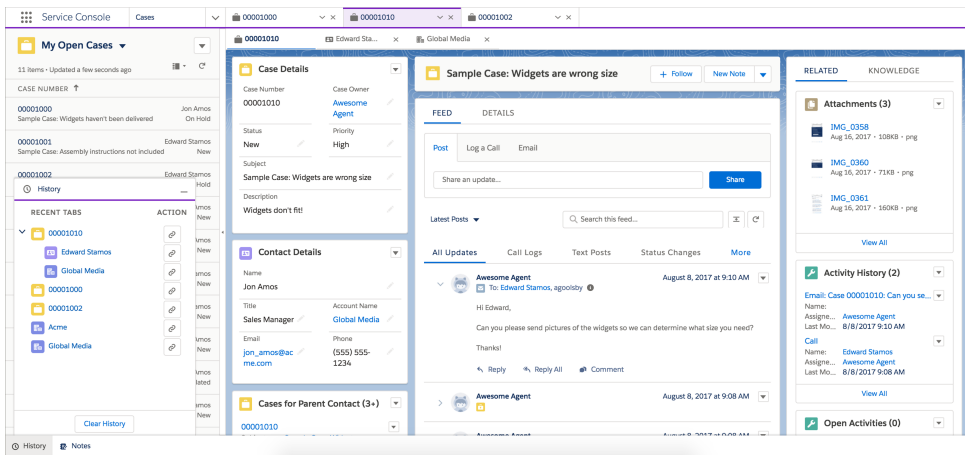
## Chapter 2 What Makes Lightning Experience So Special



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. Search for records within the current view.
5. Select which record type to view.
6. Columns are created based on the grouping field.
7. Change how columns are organized and summarized using Kanban settings.
8. Quickly move a record to a different column by dragging the card.
9. For opportunities, alerts tell how to keep a deal on track, for example, create a task or event.

## Console Apps

Minimize clicks and scrolling for your sales and service reps so they can quickly find, update, and create records. A Lightning console app is a tab-based workspace that lets users manage multiple records on a single screen. Take advantage of the standard Lightning console apps for Sales and Service to support your fast-paced environment.



## Other Highlights

- Rally your team around your brand and important initiatives, such as a new product launch, with a set of custom brand images and colors.
- Supercharge your productivity by personalizing the Lightning Experience navigation bar to suit the way you work. Reorder and add navigation items, such as a frequently used dashboard or list, to the navigation bar.
- 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. 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. And see the email messages 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. 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 global search, which includes top results and recent records.

## **Chapter 2** What Makes Lightning Experience So Special

As you can see, there's a lot of rad stuff that you get with Lightning Experience. Next, we'll take you through the key differences between Lightning Experience and Salesforce Classic, to help you figure out how you'll make the transition.

## CHAPTER 3 How Different Personas Benefit from Lightning Experience

So now you have a good sense of some of the exciting features that make Lightning Experience special. But at the end of the day, how do these features and other aspects of Lightning Experience actually benefit you and your users?

The short answer: Lightning Experience makes people more productive, no matter what role they play. And scout's honor, that's not just hyperbole. Customers who've made the move to Lightning Experience are seeing sizable increases in efficiency, including less time spent managing pipelines, faster win rates, better agent productivity, and faster deployments.

Let's look at the productivity benefits for each Salesforce persona.

### **Sales Users**

Lightning Experience is a streamlined, action-oriented environment that promotes an efficient sales workflow. In every context, pages focus on the most important information first. Workspaces make your sales process clear, supported by best practices at every step. Separate tabs for details, related information, and collaboration bring order to pages and reduce the amount of scrolling to find things. Actions like taking notes, making calls, and sending emails are right where they're needed, in context. And baked-in intelligence, from news and insights to Salesforce Einstein, tees up teams to close more deals in less time.

### **Service Users**

With Service Cloud for Lightning Experience, service agents are on the case resolution fast track. The Lightning Service Console is a high-productivity workspace with a single view of every customer so service agents don't have to waste time toggling between multiple apps. The console guides agents through your case management process, and makes every step faster to complete with efficiencies like split view, drag-and-drop attachments, and case detail hovers. Add Lightning Knowledge to the mix and agents have the tools to find and share the right information, attach articles, and get to closed cases faster.

### Admins

With the Lightning framework, admins can create and customize pages faster, without having to start from scratch each time. Don't let the term *framework* scare you off. Admins can work their magic using the framework's declarative tools that are available from Setup in Lightning Experience.

The building blocks that make up Lightning Experience are Lightning components. These components are reusable throughout the interface so it's easy to create just what's needed. We provide a bunch of pre-built components to get started, and an intuitive visual tool called Lightning App Builder. Admins can add, remove, and rearrange components on the home page or record pages with drag-and-drop ease. And beyond page layout decisions, admins can control how components behave, configuring the same component to use different properties in different locations...even on the same page. This flexibility significantly speeds up customization work.

### Developers

The Lightning framework is a productivity boon for developers too. It's fast to create custom Lightning components using the building blocks we include in the framework. And that's just the start. The framework provides many other tools to help developers. For example, the Salesforce Lightning Design System is a complete styling library so developers don't have to write their own HTML or CSS. Lightning Data Service consolidates data access code in reusable components, eliminating the need to write controller code. That's just scratching the surface. But the added efficiency for developers is obvious: all this reduced complexity means that developers can code faster and spend less time on performance and quality testing.



# CHAPTER 4 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?](#)
- [Evaluate Your Org with the Lightning Experience Readiness Check](#)
- [Preview Your Org in Lightning Experience](#)

Lightning Experience is a modern user interface that helps your sales reps sell faster and your service reps support customers more productively.

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
<ul style="list-style-type: none"><li>• Group Edition</li><li>• Essentials Edition</li><li>• Professional Edition</li><li>• Enterprise Edition</li><li>• Performance Edition</li><li>• Unlimited Edition</li><li>• Developer Edition</li></ul>	<ul style="list-style-type: none"><li>• Personal Edition</li><li>• Contact Manager Edition</li><li>• Database.com Edition</li><li>• Chatter.com Edition</li></ul>

**Table 2: Salesforce User Licenses**

Supported Licenses	Unsupported Licenses
<ul style="list-style-type: none"><li>• Salesforce</li><li>• All Salesforce Platform and Lightning Platform (excluding Force.com - Free)</li><li>• Identity User</li><li>• Company (Employee) Community</li><li>• Chatter External</li><li>• Chatter Plus</li><li>• Chatter Free</li></ul>	<ul style="list-style-type: none"><li>• Force.com - Free</li><li>• Work.com</li><li>• Database.com</li><li>• Content Only</li><li>• Ideas Only</li><li>• Knowledge Only</li><li>• Site.com Only</li></ul>

 **Important:** To learn about browser options and limitations for Lightning Experience, see [Supported Browsers for Lightning Experience](#) in the Salesforce Help.

If your org still uses Internet Explorer 11, be aware that we're in the process of retiring support for this browser. There are [significant performance issues](#) when running Lightning Experience in IE11. Even if you opt into the extended support option for IE11, we recommend migrating your users to an alternate modern browser like Google Chrome or Microsoft Edge as soon as possible.

## Comparing Lightning Experience and Salesforce Classic

Understanding what you get in the new interface is a big part of the decision to switch to Lightning Experience. We give you some highlights of how Lightning Experience compares with what you're used to in Salesforce Classic, including features that are available in the new interface only.

 **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 data for:	✓	✓
<ul style="list-style-type: none"> <li>• Accounts</li> <li>• Campaigns</li> <li>• Contacts</li> <li>• Contracts</li> <li>• Leads</li> <li>• Opportunities</li> <li>• Orders</li> <li>• Person Accounts</li> <li>• Price Books</li> <li>• Products</li> <li>• Quotes</li> </ul>		
Calendar, Events, Tasks	✓	✓
Service data for:	✓	✓
<ul style="list-style-type: none"> <li>• Assets</li> <li>• Basic Cases</li> <li>• Lightning Knowledge</li> <li>• Work Orders</li> </ul>		
Custom objects	✓	✓

## Chapter 4 Understanding Lightning Experience and Salesforce Classic

Feature	Lightning Experience	Salesforce Classic
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	✓	
News	✓	
Key deals	✓	
Tasks	✓	✓
Events <sup>1</sup>	✓	✓
Calendar <sup>1</sup>	✓	✓
Feed and publisher	✓	✓
Items to approve	✓	✓
Customizable dashboards	✓	✓
Customizable home page components	✓	✓

<sup>1</sup> In Lightning Experience, Home show 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 and service reps get a layout designed to help them find the information they need, fast.

<b>Feature</b>	<b>Lightning Experience</b>	<b>Salesforce Classic</b>
News	✓	
Path	✓	
Activity timeline	✓	
Contacts to Multiple Accounts	✓	✓
Account Logos	✓	
Automated Account Fields	✓	
Notes	✓	
“Reference” page layout	✓	
Related lists	✓	✓
Integrated email and templates	✓	✓
Twitter highlights	✓	✓
Person accounts	✓	✓
Account teams	✓	✓
Account hierarchy	✓	✓
Contact hierarchy	✓	✓
Find and merge duplicate accounts and contacts	✓	✓
Matching and duplicate rules for business accounts, person accounts, and contacts	✓	✓

## 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	✓	
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	✓	
Notes	✓	
Change lead owner to a queue	✓	✓

Feature	Lightning Experience	Salesforce Classic
Integrated email and templates	✓	✓
Lead conversion customization via the API	✓	✓
Find and merge duplicate leads	✓	✓
Matching and duplicate rules for leads	✓	✓

## Kanban View

Sales reps can use the Kanban view, a visualization tool, 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. Kanban is available on opportunities, leads, tasks, and many other objects.

Feature	Lightning Experience	Salesforce Classic
Charts	✓	
Drag and drop	✓	
Intelligent Alerts	✓	
Type-ahead list view search	✓	
Sharing settings <sup>1</sup>	✓	

<sup>1</sup> 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
Data.com Prospector: Search for companies and import them as accounts	<p>Accessed via the Get More Accounts button on the Accounts list view. You can filter on the following fields:</p> <ul style="list-style-type: none"> <li>• Annual Revenue</li> <li>• Company or Website</li> <li>• Country or Territory</li> <li>• Location Type</li> <li>• State</li> </ul> <p>To filter on other fields, use a <a href="#">web tab</a>.</p> <p>See <a href="#">Get More Accounts Using Data.com in Lightning Experience</a>.</p>	<p>Accessed via the Data.com tab or Search Global Ultimate DUNS button on D&amp;B Company records.</p>
Data.com Prospector: Search for contacts and import them as contacts	<p>Users can search for a contact when it's related to an existing account. On the account detail page, click the Get More Contacts button, and use the search box and filters.</p> <p>To search for other contacts, use a <a href="#">web tab</a>.</p> <p>See <a href="#">Prospect for Key Contacts in Your Accounts Using Lightning Experience</a>.</p>	<p>Accessed via the Data.com tab or Get Contacts button on accounts.</p> <p>See <a href="#">Prospect for Key Contacts in Your Accounts Using Salesforce Classic</a>.</p>
Data.com Prospector: Search for contacts and import them as leads	<a href="#">Web tab</a> required.	Accessed via the Data.com tab.
Data.com Prospector: Export companies or contacts to a .csv file	<a href="#">Web tab</a> required.	Accessed via the Data.com tab.



## Chapter 4 Understanding Lightning Experience and Salesforce Classic

Feature	Lightning Experience	Salesforce Classic
Data.com Prospector: See key company insights on accounts	Accessed via the Get More Insights button on accounts.  See <a href="#">Prospecting Insights in Lightning Experience</a> .	Accessed via the Prospecting Insights button on accounts.
Data.com Prospector: See key company insights on opportunities	Accessed via the Get More Insights button on opportunities.  See <a href="#">Prospecting Insights in Lightning Experience</a> .	Not available.
Data.com Prospector: Prospect for accounts within the same corporate hierarchy	On accounts and opportunities, accessed via the Get More Insights button (Data.com Insights component) or via the Company Hierarchy component.  See <a href="#">Prospecting Insights in Lightning Experience</a> .	Accessed via the Company Hierarchy button on accounts.
Data.com Clean: Administrative settings	Not available. Use Salesforce Classic to administer Clean jobs and configure Clean settings.	Administer Clean jobs, and configure Clean settings.
Data.com Clean: Clean accounts with company data	Bulk cleaning: Clean jobs for accounts.  Manual cleaning: Available using the Company Info for Accounts data integration rule (Premium Clean required).  See <a href="#">Update Accounts and Leads with Key Company Information</a> .	Bulk cleaning: Clean jobs for accounts.  Manual cleaning: Clean button on accounts.

## Chapter 4 Understanding Lightning Experience and Salesforce Classic

Feature	Lightning Experience	Salesforce Classic
Data.com Clean: Clean contacts with contact data	<p>Bulk cleaning: Clean jobs for contacts.</p> <p>Manual cleaning: Available using <a href="#">Data.com Contact Clean</a>, a free component from Salesforce Labs on AppExchange.</p>	<p>Bulk cleaning: Clean jobs for contacts.</p> <p>Manual cleaning: Clean button on contacts.</p>
Data.com Clean: Clean leads with contact data	<p>Bulk cleaning: Clean jobs for leads.</p> <p>Manual cleaning: Not available.</p>	<p>Bulk cleaning: Clean jobs for leads.</p> <p>Manual cleaning: Clean button on leads.</p>
Data.com Clean: Clean leads with company data	<p>Available using the Company Info for Leads data integration rule (Premium Clean required).</p> <p>See <a href="#">Update Accounts and Leads with Key Company Information</a>.</p>	Not available.
Data.com Clean: View and refresh D&B company records	Accessible via a link on an account or lead. A Salesforce Classic tab opens. The tab doesn't interrupt the user's Lightning Experience session.	Accessible via a link on accounts or leads and from the D&B Companies tab.

## Collaborative Forecasts

Feature	Lightning Experience	Salesforce Classic
Forecasting on:	✔	✔
<ul style="list-style-type: none"> <li>• The opportunity Amount field</li> <li>• Opportunity splits</li> <li>• Overlay splits</li> <li>• Custom opportunity fields</li> <li>• Product families</li> </ul>		

## Chapter 4 Understanding Lightning Experience and Salesforce Classic

Feature	Lightning Experience	Salesforce Classic
Revenue and quantity forecasts	✓	✓
Product and schedule date forecasts (for the opportunity revenue and quantity forecast types)	✓	
Territory forecasts	✓	
Forecast sharing	✓	
Manager and owner adjustments	✓	✓
Cumulative forecast rollups	✓	✓
Support for multiple currencies	✓	✓
Quotas	✓	✓
Forecast hierarchy	✓	✓
Choose the product families to forecast on and the order they appear in (from Setup, if product family forecasts is selected)	✓	
Forecasts grid organized first by period, product family, and then user to show product family forecast totals	✓	
Forecasts grid expansion remains the same during each session	✓	
Timestamp showing refresh time and date	✓	
All-zero rows hidden for all forecast types	✓	
Adjust your own product family forecasts	✓	
Opportunity split details in the opportunities list		✓
Adjustable column width	✓	✓
Quota attainment information	✓	✓

## Chapter 4 Understanding Lightning Experience and Salesforce Classic

Feature	Lightning Experience	Salesforce Classic
Jump to the forecasts of users you have access to view	✓	✓
Forecasts setup	✓	✓

## Campaigns

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	✓	✓
Add campaign members from reports	✓	✓

## Salesforce Einstein

Feature	Lightning Experience	Salesforce Classic
<b>Analytics</b>		
Einstein Discovery	✓	✓
<b>Chatter</b>		
Einstein Recommendations	✓	✓
<b>Community</b>		
Einstein Related Articles	✓	✓

<b>Feature</b>	<b>Lightning Experience</b>	<b>Salesforce Classic</b>
Einstein Feed Search	✓	✓
<b>Development</b>		
Einstein Language	✓	✓
Einstein Vision	✓	✓
<b>Sales</b>		
Einstein Activity Capture	✓	
Einstein Lead Scoring	✓	✓
Einstein Opportunity Scoring	✓	✓
Einstein Forecasting	✓	✓
Einstein Opportunity Insights	✓	
Einstein Account Insights	✓	
Einstein Automated Contacts	✓	
<b>Service</b>		
Einstein Bots	✓	

## Customer Service

<b>Feature</b>	<b>Lightning Experience</b>	<b>Salesforce Classic</b>
Assets	✓	✓
Case Assignment Rules	✓	✓
Case Comments	✓	✓
Case Feed Filters	✓	✓
Case Teams	✓	✓
Console <sup>1</sup>	✓	✓

## Chapter 4 Understanding Lightning Experience and Salesforce Classic

Feature	Lightning Experience	Salesforce Classic
Einstein Bots	✓	
Entitlements <sup>2</sup>	✓	✓
Field Service Lightning	✓	✓
Knowledge <sup>3</sup>	✓	✓
Lightning Flow for Service	✓	
Live Agent	✓	✓
LiveMessage	✓	
Macros	✓	✓
Milestones—Case	✓	✓
Milestones—Object	✓	✓
Omni-Channel	✓	✓
Open CTI	✓	✓
Products	✓	✓
Quick Text	✓	✓
Service Contracts	✓	✓
Service Setup Flows for:	✓	
<ul style="list-style-type: none"> <li>• Communities</li> <li>• Email-to-Case</li> <li>• Facebook and Twitter</li> <li>• Help Center</li> <li>• Knowledge</li> <li>• Live Agent</li> <li>• LiveMessage</li> <li>• Omni-Channel</li> </ul>		
Snap-ins for Web <sup>4</sup>	✓	✓

<b>Feature</b>	<b>Lightning Experience</b>	<b>Salesforce Classic</b>
Social Customer Service	✓	✓
Solutions		✓
SOS		✓

<sup>1</sup> The Service Console in Lightning Experience doesn't yet have full parity with the Service Console in Salesforce Classic. [Learn more.](#)

<sup>2</sup> There are some limitations using Entitlements in Lightning Experience. [Learn more.](#)

<sup>3</sup> Enabling Lightning Knowledge changes your org's data model to use record types rather than article types. To move from Salesforce Classic Knowledge to Lightning Knowledge, use the Lightning Knowledge Migration Tool (Beta). Test your data migration in a sandbox or trial org before enabling Lightning Knowledge in your production org. After you enable Lightning Knowledge, you can't disable it. Lightning Knowledge doesn't yet have full parity with Knowledge in Salesforce Classic. [Learn more.](#)

<sup>4</sup> Snap-ins setup is available in Lightning Experience only. However, the snap-in itself can go anywhere on your website.

## Reports

Sales and service 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.

<b>Feature</b>	<b>Lightning Experience</b>	<b>Salesforce Classic</b>
Enhanced report charts	✓	
Hide totals and subgroups from report view	✓	
Interactive filters when viewing reports	✓	
Report Builder	✓	✓
Currency selector in the Report Builder	✓	

## Chapter 4 Understanding Lightning Experience and Salesforce Classic

Feature	Lightning Experience	Salesforce Classic
<b>Product standard filter for the Opportunities with Contact Roles and Products</b> report type in the Report Builder		✓
<b>Select a campaign</b> option on the <b>Show Me</b> filter for Campaign report types in the Report Builder		✓
Bucket fields	✓	✓
Custom summary formulas	✓	✓
Matrix, Summary, Tabular report formats	✓	✓
Stacked summaries in the Matrix report format		✓
Column (non-stacked) summaries in the Matrix report format	✓	
Role hierarchy filters	✓	✓
Create report folders <sup>1</sup>	✓	✓
Joined reports <sup>2</sup>	✓	✓
Pie charts		✓
Schedule report refreshes <sup>3</sup>	✓	✓
Follow reports		✓
Report notifications		✓
Export reports	✓	✓
Historical trend reporting		✓
Conditional formatting <sup>4</sup>		✓

<sup>1</sup> Folder sharing setup isn't available in Lightning Experience. But folders created in Salesforce Classic inherit all assigned permissions when the folders are viewed in Lightning Experience.

<sup>2</sup> Joined reports are beta in Lightning Experience.



<sup>3</sup>Available in Lightning Experience by subscribing to a report.

<sup>4</sup>You can configure conditional highlighting in Lightning Experience using the Salesforce Classic report builder, but not using the Lightning Experience report builder. Though configurable, conditional highlighting doesn't display on reports in Lightning Experience.

## Dashboards

Feature	Lightning Experience	Salesforce Classic
Display more than 3 columns	✓	
Dashboard Builder	✓	✓
View filtered dashboards	✓	✓
View dynamic dashboards	✓	✓
Schedule dashboard refreshes <sup>1</sup>		✓
Post dashboard components to feeds	✓	✓
Follow dashboards	✓	✓
Visualforce components		✓

<sup>1</sup> Users can't schedule dashboards in Lightning Experience, but dashboards scheduled in Salesforce Classic run as expected in Lightning Experience. Instead of scheduling dashboards in Lightning Experience, subscribe to them.

## 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	✓	

## Chapter 4 Understanding Lightning Experience and Salesforce Classic

Feature	Lightning Experience	Salesforce Classic
View records visually (Kanban)	✓	
Create and edit lists	✓	✓
Sortable columns	✓	✓
Resizable columns	✓	✓
Sharing Settings <sup>1</sup>	✓	✓
Filter logic	✓	✓
Inline editing	✓	✓

<sup>1</sup> 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
Custom org brand images and colors	✓	
Custom branding of Salesforce apps	✓	
Personalized Navigation Bar	✓	✓
Favorites	✓	
Activities-related composer windows	✓	
Create and edit records	✓	✓
Inline editing record detail fields	✓	✓
Inline editing in lists	✓	✓
Customizable Forecasting		✓
Enterprise Territory Management	✓	✓
Original Territory Management <sup>1</sup>	✓	✓

Feature	Lightning Experience	Salesforce Classic
Salesforce Communities <sup>2</sup>	✓	✓
Partner Portals		✓
Work.com		✓

<sup>2</sup>To create or disable community users, log in as a community user, and use delegated administration, switch to Salesforce Classic.

## Other Features That Aren't Supported

There are some other features that aren't available or don't work in Lightning Experience. We want to be transparent with you about them, so you know about customizations you may need to update before enabling 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 and links that use a JavaScript content source aren't supported in Lightning Experience. Check out these resources for migrating your JavaScript custom button and link functionality to Lightning-friendly options.

## 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.

## 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 evaluation process.

## Evaluate Your Org with the Lightning Experience Readiness Check

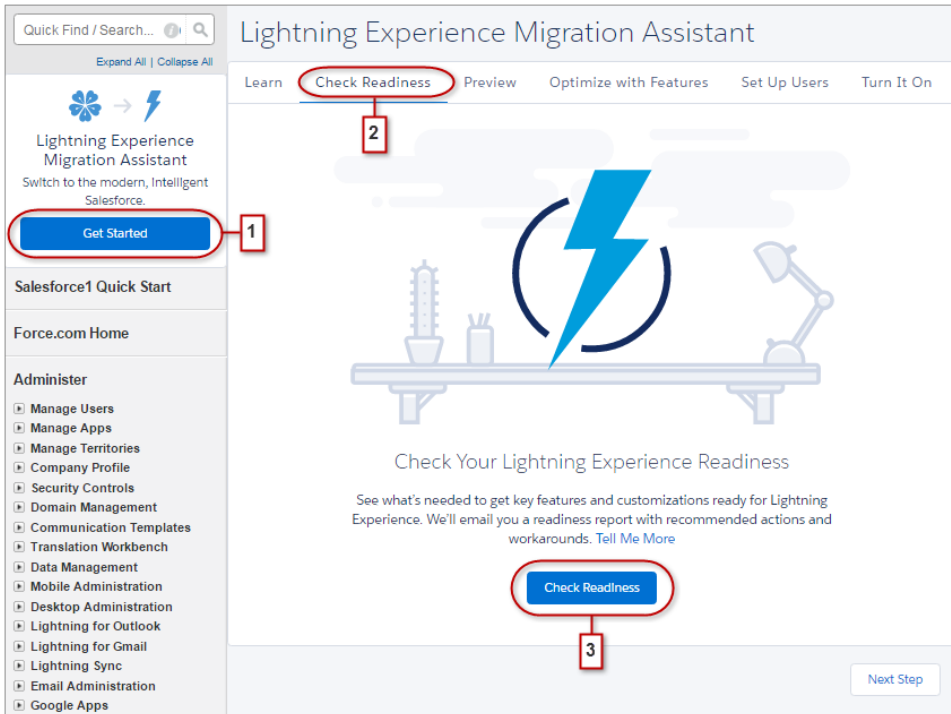
---

To evaluate whether some of the key features and customizations you’ve implemented in Salesforce Classic are ready for Lightning Experience, kick off the Lightning Experience Readiness Check. That way, you can learn which are ready now for Lightning Experience and which may need your attention before you switch over all your users.

You’ll get a personalized Readiness Report, with an assessment of the features and customizations evaluated by the Readiness Check and recommendations on your next steps. Plus you’ll be eligible for a free consultation with a Lightning expert to go over the details in your report. The Readiness Check emails the report to you and also saves it to Salesforce Files.

Curious what exactly is covered in the report? Check out [What Does the Lightning Experience Readiness Check Evaluate?](#) in the Salesforce Help.

Run the Readiness Check from the Lightning Experience Migration Assistant. From Setup in Salesforce Classic, click **Get Started** in the Migration Assistant tile (1). On the Check Readiness tab in the Migration Assistant (2), click **Check Readiness** (3).



We've also put together an [enablement pack](#) with sample documents and templates, including a gap analysis checklist. You can use this checklist to expand your evaluation to features and customizations that aren't covered by the Readiness Check.

## 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. And helpful hint, the Preview is a great way to review the state of features and customizations that aren't evaluated by the Lightning Experience Readiness Check.

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.

## Chapter 4 Understanding Lightning Experience and Salesforce Classic

To get started, from Setup in Salesforce Classic, click **Get Started** in the Lightning Experience Migration Assistant tile at the top of the menu. On the Preview tab in the Migration Assistant, click **Preview** to dive in.

# CHAPTER 5 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 235 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 optimized to help your sales reps sell faster and help your service reps support customers faster.

### **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. Adding the Files related list lets users upload files to records, see files associated with records, and quickly page through files in the visually rich preview player.

### **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. Adding the Notes related list lets your users create and read notes directly from their records.

### **Path**

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

### **Duplicate Management**

To alert users when they're about to create a duplicate record, create duplicate rules. Duplicate rules establish 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, consider whether you 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 subdomain with the Salesforce My Domain feature. A My Domain subdomain helps you manage login and authentication. Set up a My Domain subdomain to use Lightning components in Lightning component tabs, Lightning pages, the Lightning App Builder, and standalone apps. To set up a subdomain in your production org, see the [User Authentication](#) module. My Domain is already turned on in all Trailhead Playgrounds. Do not attempt to turn on My Domain, or change its settings, in your Trailhead Playground.

### Post Pinning

Pin up to three posts to the top of group and topics feeds. Give the best visibility possible to high-impact posts that can otherwise get buried in a busy feed.

### Code Snippets

Enter syntax-highlighted code samples through the Chatter publisher.

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. Just look for the Lightning Experience Migration Assistant tile at the top of the Setup menu.

## Set Up Users for Lightning Experience

---

Next up, make sure that the right users can access Lightning Experience when it's turned on. You can fine-tune access with permission sets and custom profiles. Meaning you can do a phased rollout, starting with a small pilot group or a specific team of users who can benefit from the new interface. Or go for it and set your entire organization loose. You can also control who immediately switches to the new interface, who stays in Salesforce Classic until they switch themselves, and who can't switch back to the classic interface. The power is yours.

## Chapter 5 Your Lightning Experience Rollout Strategy

 **Note:** Okay, if you use Essentials or 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" in Essentials Edition and Group Edition—all your users get access when Lightning Experience is enabled. But you can control who switches immediately to the new interface and who remains in the classic interface until they're ready to switch themselves.

Users require the Lightning Experience User permission to access Lightning Experience. By default, this permission is automatically enabled for all users with a standard Salesforce profile. Custom profiles don't automatically include the permission.

So first things first, do you have users with standard profiles who you don't want accessing Lightning Experience yet? Move these users to custom profiles that don't include the Lightning Experience User permission.

For users assigned to custom profiles, decide who gets Lightning Experience access now and who you'll set up later. 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.
- Interested in rolling out Lightning Experience to everyone assigned 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 a Data Loader ninja, you can save time by updating many profiles at once. But remember, with great power comes great responsibility. Take care you don't inadvertently enable or disable other features.

If you're limiting Lightning Experience to a subset of your users, try to 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 [Considerations for Switching Between Lightning Experience and Salesforce Classic](#) for more details.

## Decide Who Immediately Switches to Lightning Experience and Who Can't Switch Back to Salesforce Classic

You obviously want the users you've set up for Lightning Experience to start working in the new interface sooner than later. There are several options you can use to encourage adoption of Lightning Experience. Choose what's right for each user.

- By default, Lightning Experience-enabled users stay in Salesforce Classic and automatically get the Switcher, so they can switch themselves when they're ready. This may be the best option if you want to enable Lightning Experience now but your team isn't fully trained yet or it's not a good time to introduce a major change.
  - For users who are trained and ready to go, make Lightning Experience their default experience. These users also get the Switcher, so they can return to Salesforce Classic if needed. See [Switch Users to Lightning Experience](#).
  - Encourage users to stay in Lightning Experience by automatically switching them to the new interface on a daily or weekly schedule. These users can return to Salesforce Classic, but they'll keep getting switched back to Lightning Experience at the next scheduled interval. See [Encourage Users to Stay in Lightning Experience](#).
  - When you're ready for users to dive into Lightning Experience without looking back, you can move them to the new interface without the option to switch back to Salesforce Classic. See [Make Lightning Experience the Only Experience](#).
-  **Note:** After Lightning Experience is enabled, any new users you add automatically get Lightning Experience as their default experience if they also have the Lightning Experience User permission—whether from a standard profile or a custom profile that includes the user permission. New users also get the Switcher so they can access Salesforce Classic, unless you opt to restrict them to Lightning Experience.

## Turn Lightning Experience On!

---

With the right users set up for Lightning Experience, you're ready to go live. Fortunately, it's easy to turn on Lightning Experience using the Lightning Experience Migration Assistant.

From Setup in Salesforce Classic, click **Get Started** in the Migration Assistant tile at the top of the menu.

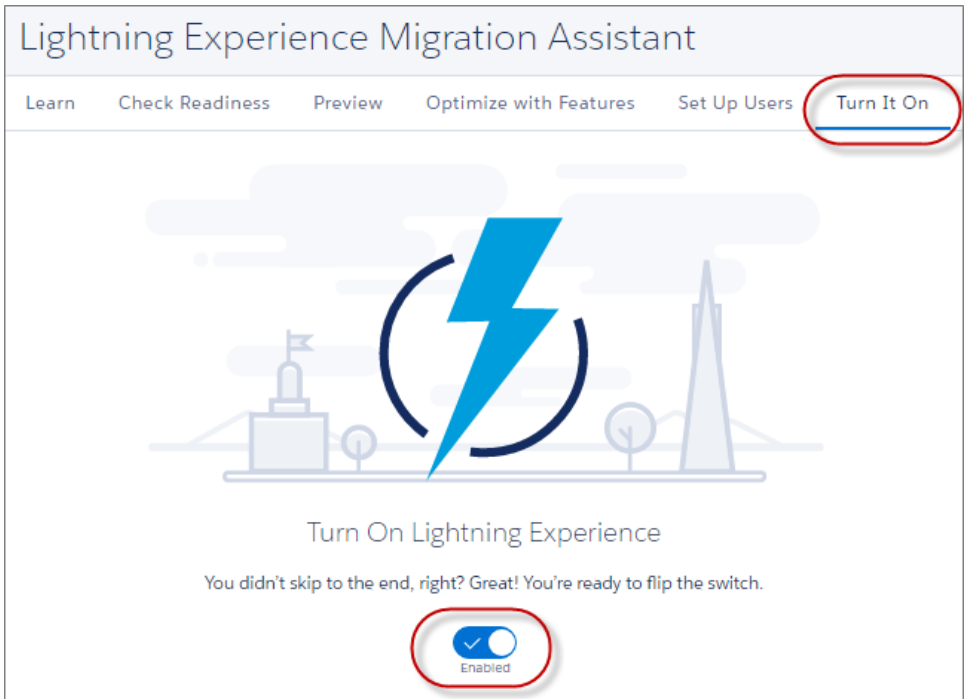
## Chapter 5 Your Lightning Experience Rollout Strategy

The screenshot shows the Salesforce Lightning Experience dashboard. At the top, there is a search bar and navigation links for Admin User, Setup, Help & Training, and Sales. Below this is a horizontal menu with tabs for Home, Chatter, Leads, Accounts, Contacts, Opportunities, Reports, Dashboards, Documents, Products, and Forecasts. A 'Quick Find / Search...' bar is located below the menu. The main content area is divided into several sections:

- Lightning Experience Migration Assistant:** This section is highlighted with a red box. It features a lightning bolt icon and the text: "Lightning Experience Migration Assistant. Switch to the modern, Intelligent Salesforce." Below this is a blue "Get Started" button.
- Getting Started:** This section includes a "Build App" card with a hammer icon, a description: "Generate a basic app with just one step, and then easily extend that app with clicks or code." and an "Add App" button. It also has a "Learn More" section with links to "Next Steps", "Force.com Workbook", "Force.com Fundamentals", and "Download SalesforceA, the mobile app for admins: iOS | Android".
- Recent Items:** A table listing recent items with columns for Name, Type, and Object.
- Trailhead:** A green banner with the text "the fun way to learn Salesforce" and a "VIEW FREE TUTORIALS" button.
- Browse Docs:** A section with a code icon and the text "Explore documentation and API reference guides."
- Ask Questions:** A section with a speech bubble icon and the text "Get help when you need it from Salesforce & community experts."

Name	Type	Object
Karen Adams	User	
Executive User	Profile	
Chatter Expert	User	
Custom-baby	Profile	
Tam's custom profile	Profile	
DataAssessment	Custom App	

On the Turn It On tab in the Migration Assistant, click the button to switch it to **Enabled**.



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.

## Switching Between Lightning Experience and Salesforce Classic

---

Ideally, Lightning Experience meets the needs of everyone in your org. But just in case someone needs access to a feature or tool back in the classic interface, we provide the Switcher. By default, users enabled for Lightning Experience automatically get the Switcher. Which they can use to...wait for it...switch between Lightning Experience and Salesforce Classic whenever they need.

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.

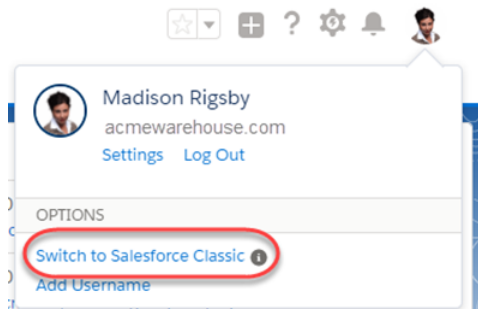
## Chapter 5 Your Lightning Experience Rollout Strategy

How do you get to the Switcher?

In the Salesforce Classic header, click **Switch to Lightning Experience**. (The Switcher link is also available from a user's profile menu.)



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




## A Few Switcher Gotchas

Switching between Lightning Experience and Salesforce Classic affects the underlying URL routing logic and can lead to some unexpected results when links are resolved. There can be snags with bookmarks and sharing links in emails or Chatter posts—especially if only a subset of your users is enabled for Lightning Experience.

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. Remember, you can roll out Lightning Experience to a specific set of users using profiles or permission sets.





# WHAT'S NEW IN LIGHTNING EXPERIENCE

## CHAPTER 6 Navigation and Setup in Lightning Experience

In this chapter ...

- [The Lightning Experience Navigation Bar](#)
- [The App Launcher in Lightning Experience](#)
- [Search for Records](#)
- [Lightning Experience Help Menu](#)
- [Meet the New and Improved Setup](#)
- [Object Manager](#)
- [Service Setup Flows](#)

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?

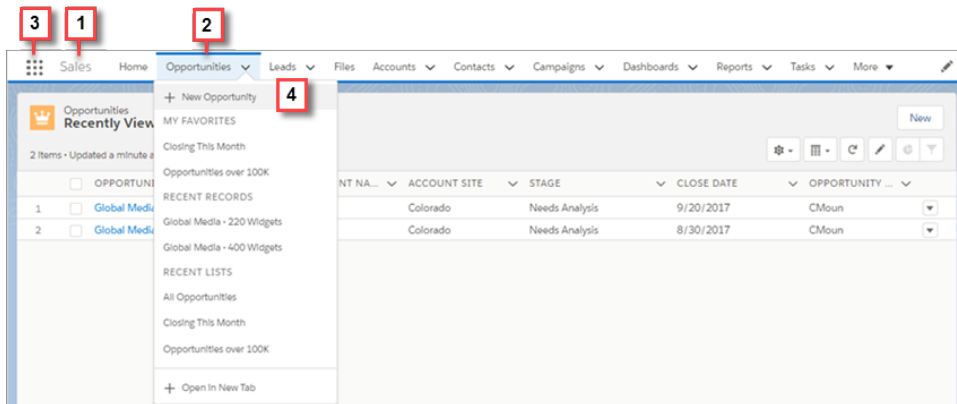
## The Lightning Experience Navigation Bar

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
- Personalize the navigation bar to suit the unique way they work

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.

## 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.

## How can users personalize the navigation bar?

Personalized navigation in Lightning Experience is similar to customized tab sets in Salesforce Classic but better. In Lightning Experience, the navigation bar can contain more than just object-level items, like Accounts. You can add granular items, like a dashboard, list, or record. Let's look at the different ways users edit the navigation bar.

You can drag items on the navigation bar to change their order.



**Note:** The asterisk to the left of the item name indicates that it's a temporary tab. A temporary tab opens when you open an item that doesn't have a parent object already in the navigation bar. The temporary tab is removed from the nav bar when you close it, log out of Salesforce, or switch to a different app.

Want to make several changes to the navigation bar? Click the pencil icon on the navigation bar. From the enhanced editor:

- Reorder the items already in your navigation bar.
- Rename items you've added. A pencil icon is next to the items that you can rename
- Add items to the navigation bar. Click **Add More Items**. Search through your favorites or all available items in your org, and choose what to add. After you make your selections, you can reorder or remove items before saving your changes. You can't delete items that your admin has specified for the app.

A few considerations to keep in mind.

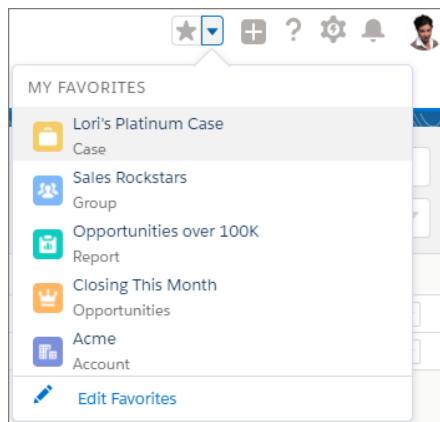
- Help users get the most out of personalized navigation by upgrading your Classic apps to Lightning apps. Users can't personalize the navigation bar of Classic apps in Lightning Experience.
- Check what's in your apps. Users can't remove the items you include in the navigation bar, and they can't personalize the navigation bar when it contains more than 50 items. For example, if you include 32 items in an app's navigation bar, users can add 18 more personal items.

## Chapter 6 Navigation and Setup in Lightning Experience

- Items that you add to an app’s navigation bar are added to the end of users’ personalized navigation bars in the order that you added them.
- When you remove an item from an app, that item remains in your users’ personalized navigation bars, and users can then delete it.
- If you don’t want your users to personalize the navigation bar for a specific app, disable personalization. From Setup in Lightning Experience, go to the **App Manager**. For the desired app, select **App Options**. Select **Disable end user personalization of nav items in this app**.
- If you don’t want your users to personalize the navigation bar for any app, disable personalization. From Setup, enter *User Interface* in the Quick Find box, then select **User Interface**. Select **Disable Navigation Bar Personalization in Lightning Experience**. Salesforce recommends disabling navigation personalization by app instead of for the entire org.
- Control if temporary tabs are created when users access items outside of the app. From Setup in Lightning Experience, go to the **App Manager**. Edit the desired app. On the **App Options** page, select **Don’t automatically create temporary tabs when users access items outside this app**.

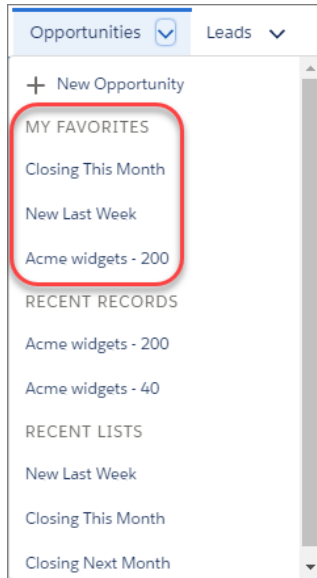
## Fast Track Navigation in Lightning Experience with Favorites

Favorites are another great way for your users to boost productivity by personalizing how they navigate. Think of the favorites star at the top of the page in Lightning Experience as a constellation in the sky. It’s always there, so your users can easily navigate to what they need no matter where they are.



Click the favorites star to add the current page—for example, opportunities closing this month—to your favorites. Saved favorites are always just a click away in the favorites list. A highlighted star means you’re on a favorite page. To remove a favorite, click the highlighted star. Your users can easily personalize their favorites lists by renaming favorites and rearranging items in the list.

If an item in the navigation bar has a dropdown menu, it automatically includes the user's top favorites for the item.



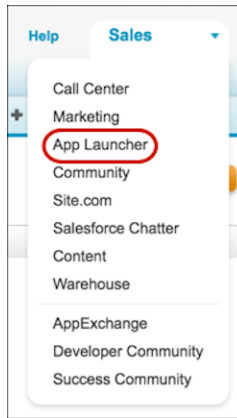
## The App Launcher in Lightning Experience


---

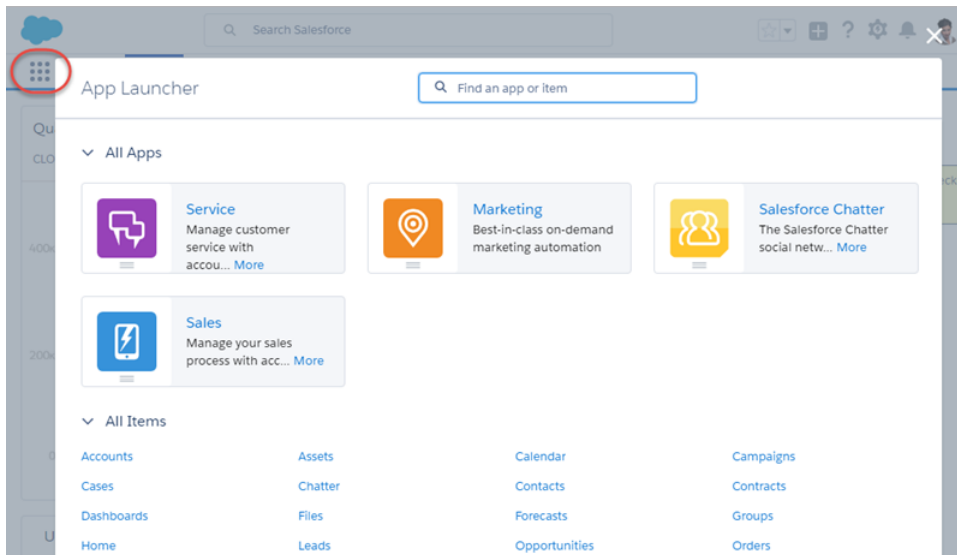
In Lightning Experience, your users switch between apps through the App Launcher. They can browse the App Launcher to find available apps. Or they can search for an app by name from the Find an app or item box at the top of the App Launcher.

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

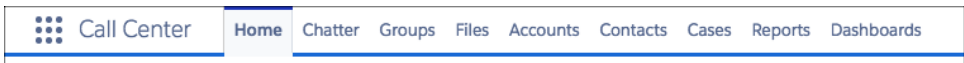
## Chapter 6 Navigation and Setup in Lightning Experience



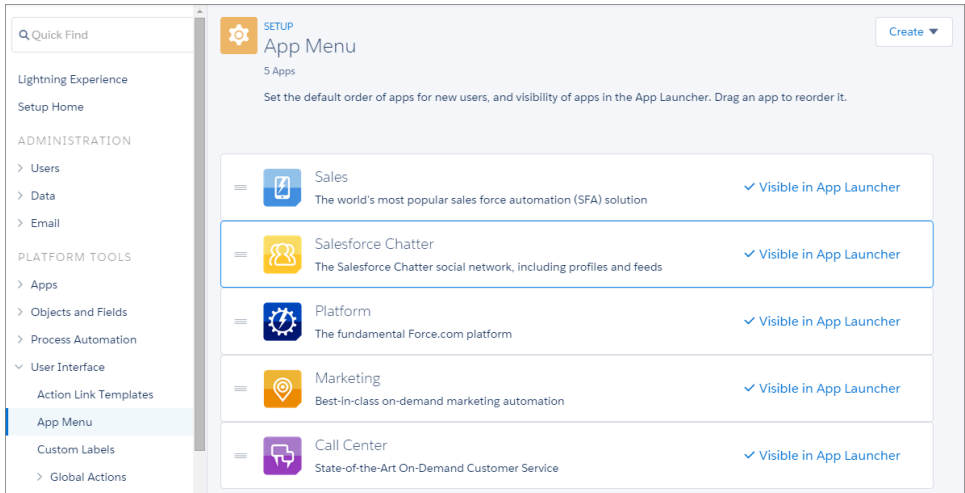
To get to the Lightning Experience App Launcher, click  from any page. Apps show up as large tiles under All Apps. Apps can include Salesforce standard apps, custom apps, and connected apps like G Suite. Other items, such as custom objects, tasks, events, and the feed, show up under All Items.



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 in the Lightning Experience App Launcher. You can also control the order in which the apps appear from the app menu.

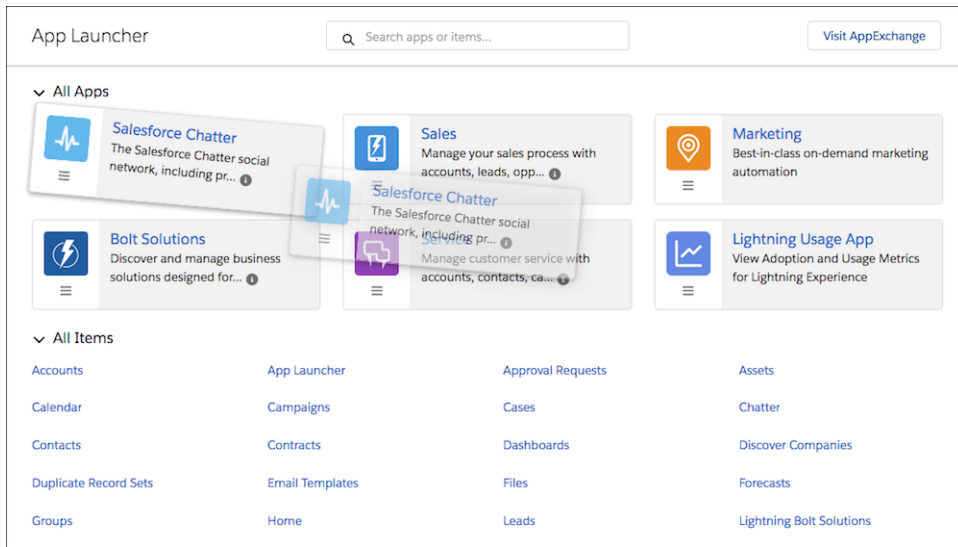


1. From Setup, enter *App Menu* in the Quick Find box, then select **App Menu**.
2. From the list of app menu items, drag the apps to change their order. Changes take effect immediately.
3. Optionally, click **Visible in App Launcher** or **Hidden in App Launcher** to show or hide individual apps from 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. Their sort order overrides the order that you set here.

## Chapter 6 Navigation and Setup in Lightning Experience

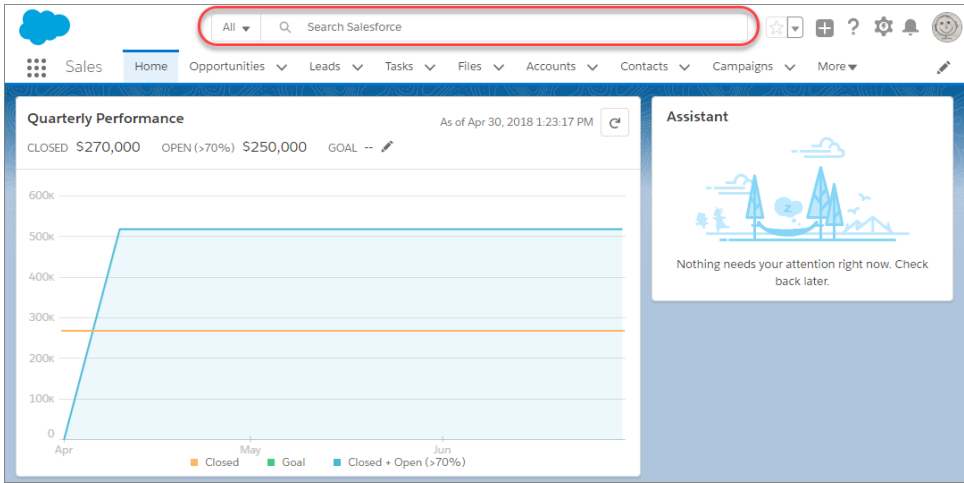


## Search for Records

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 global 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.

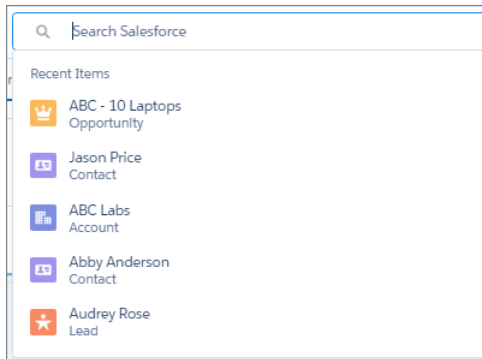
Think of search as your personal assistant—your very smart personal assistant. It anticipates your needs and helps you find what you're looking for from anywhere in the app.





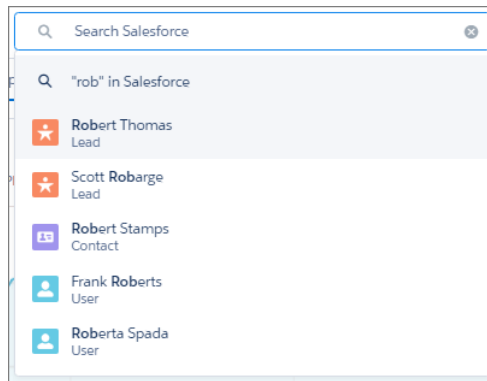
## Get Instant Results, Top Results, and More

As soon as you click into the box, search starts working for you. A list of recent items provides quick links to records you've recently visited.

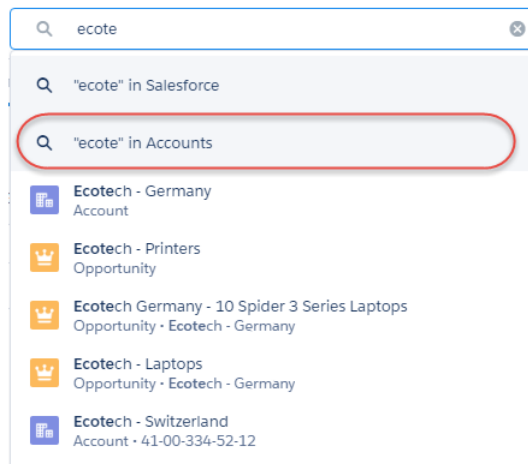


Start typing, and the list dynamically updates with matches from all searchable objects. If you see what you're looking for, select it to go right to the record.

## Chapter 6 Navigation and Setup in Lightning Experience



Don't see what you need in the dropdown list? 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 dropdown list.



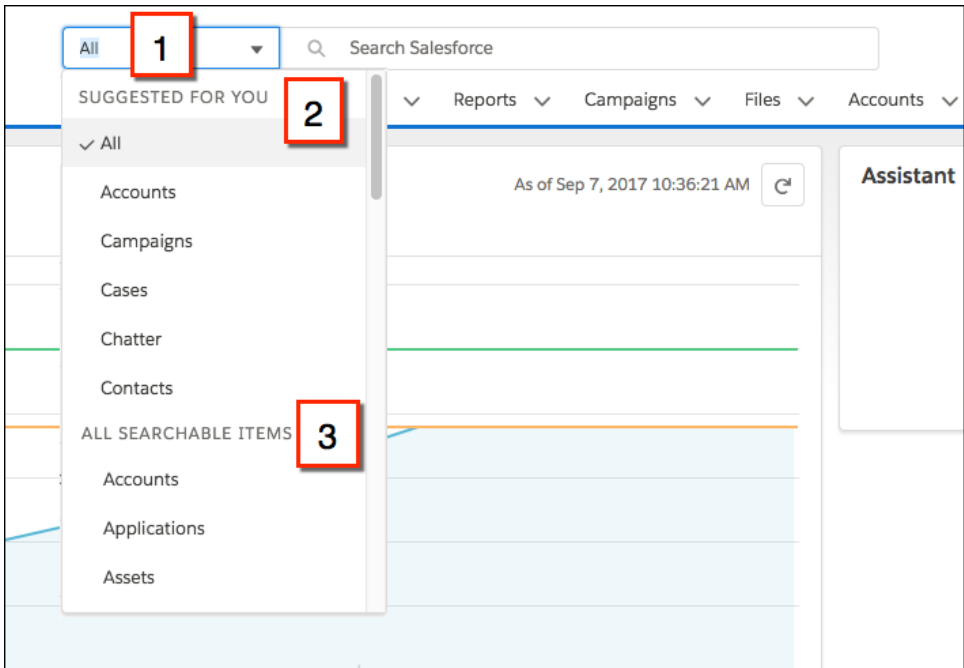
Alternatively, you can select the first option or press Enter to search across your entire organization. You'll land on the Top Results page, which shows you the most relevant results from your most frequently used objects.

You also get a glimpse of how search helps you find what you need. Features like spelling correction and synonyms return matches that are similar to your search term. You might find one that's just what you're looking for.

Know which object you want to search? Start typing the name of the object next to the global search box. You can filter the dropdown list (1), or scroll to find what you're looking for. The objects you use most frequently are at the top (2), followed by all searchable objects, listed alphabetically (3).

To select a highlighted object, click it or press Enter. As you type your search term, the instant results are limited to the selected object. Press Enter to see full search results for only that object.

To remove the filter, change the object to **All**.



## Refine Your Results

From the high-level overview of Top Results, it's easy to zoom in on what you need. See results for a specific object by selecting it on the left, under Search Results.

If you don't see an object that 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.

## Chapter 6 Navigation and Setup in Lightning Experience

The screenshot shows the Salesforce Lightning Experience search results page. The left sidebar, titled "SEARCH RESULTS", lists various object types: Top Results, Opportunities, Leads, Tasks, Files, Accounts, Contacts, Campaigns, Dashboards, Reports, Chatter, Groups, Events, People, and Cases. A red box highlights the "Opportunities" and "Leads" sections in the sidebar. The main content area displays search results for "Opportunities", "Leads", "Accounts", and "Contacts".

**Opportunities**  
5+ Results - Sorted by Relevance

OPPORTUNITY NAME	ACCOUNT NAME	STAGE	CLOSE DATE	OPPORTUNITY OWNER ALIAS
<a href="#">Acme - 100 Widgets</a>	<a href="#">Acme</a>	Proposal/Price Quote	1/18/2018	AUser
<a href="#">drill more oil</a>	<a href="#">Acme</a>	Closed Won	1/8/2018	AUser
<a href="#">Drill oil</a>	<a href="#">Acme</a>	Closed Won	1/8/2018	AUser
<a href="#">Acme</a>	<a href="#">Acme Printing</a>	Needs Analysis	8/31/2017	AUser
<a href="#">Acme - 200 Widgets</a>	<a href="#">Acme</a>	Value Proposition	5/14/2015	AUser

**Leads**  
1 Result

NAME	TITLE	COMPANY	PHONE	EMAIL	LEAD STATUS	OWNER ALIAS
<a href="#">Wros</a>		<a href="#">acme</a>			Nurturing	AUser

**Accounts**  
2 Results - Sorted by Relevance

ACCOUNT NAME	PHONE	ACCOUNT OWNER ALIAS
<a href="#">Acme</a>	(212) 555-5555	AUser
<a href="#">Acme Printing</a>		AUser

**Contacts**  
3 Results - Sorted by Relevance

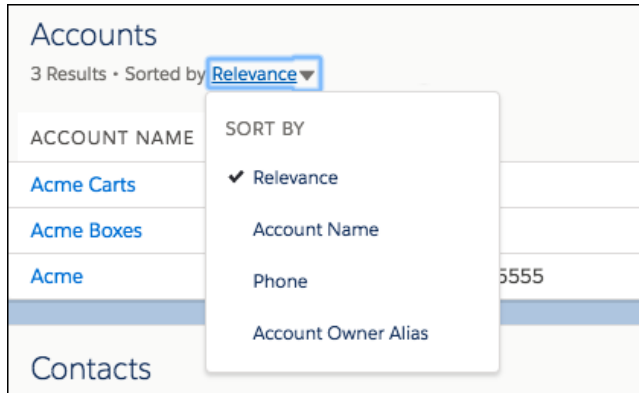
NAME	ACCOUNT NAME	PHONE	EMAIL	CONTACT OWNER ALIAS
<a href="#">Howard Jones</a>	<a href="#">Acme</a>	(212) 555-5555	<a href="mailto:info@salesforce.com">info@salesforce.com</a>	AUser
<a href="#">Leanne Tomlin</a>	<a href="#">Acme</a>	(212) 555-5555	<a href="mailto:info@salesforce.com">info@salesforce.com</a>	AUser
<a href="#">Edward Stamos</a>	<a href="#">Acme</a>	(212) 555-5555	<a href="mailto:info@salesforce.com">info@salesforce.com</a>	AUser

You can filter search results for accounts, cases, contacts, dashboards, files, Knowledge articles, leads, notes, opportunities, people, and tasks, and for custom objects. To see filtering options, click the object name in the Search Results sidebar.

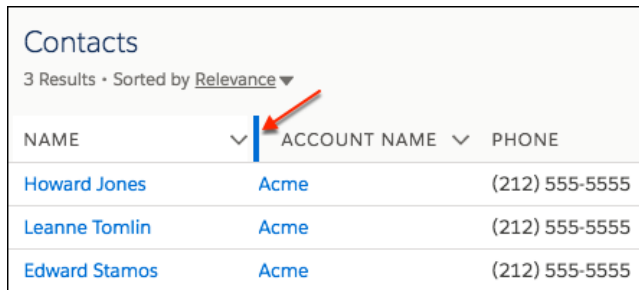
You can also sort most search results. When you first get to the search results page, you see that results are sorted by relevance. That sounds smart—and it is!—but what does it mean, exactly? Several things are considered, like:

- How unique is the search term?
- How often does the search term appear in a record?
- Do you own the record you're searching for?

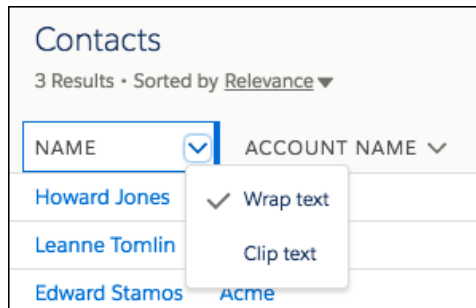
You can also sort search results by clicking column headers or using the sort dropdown menu.



You can adjust column widths on the search results page by clicking and dragging the borders in the column header.



Text wrapping is on by default, but you can change this preference for a column by clicking the down arrow in the column header.



## Can't Find What You're Searching For?

- If you get too many results, try using more—and more specific—search terms.
- Check whether the object or field is searchable.
- Make sure that 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.

For more tips, see Search for Records in the Resources section.

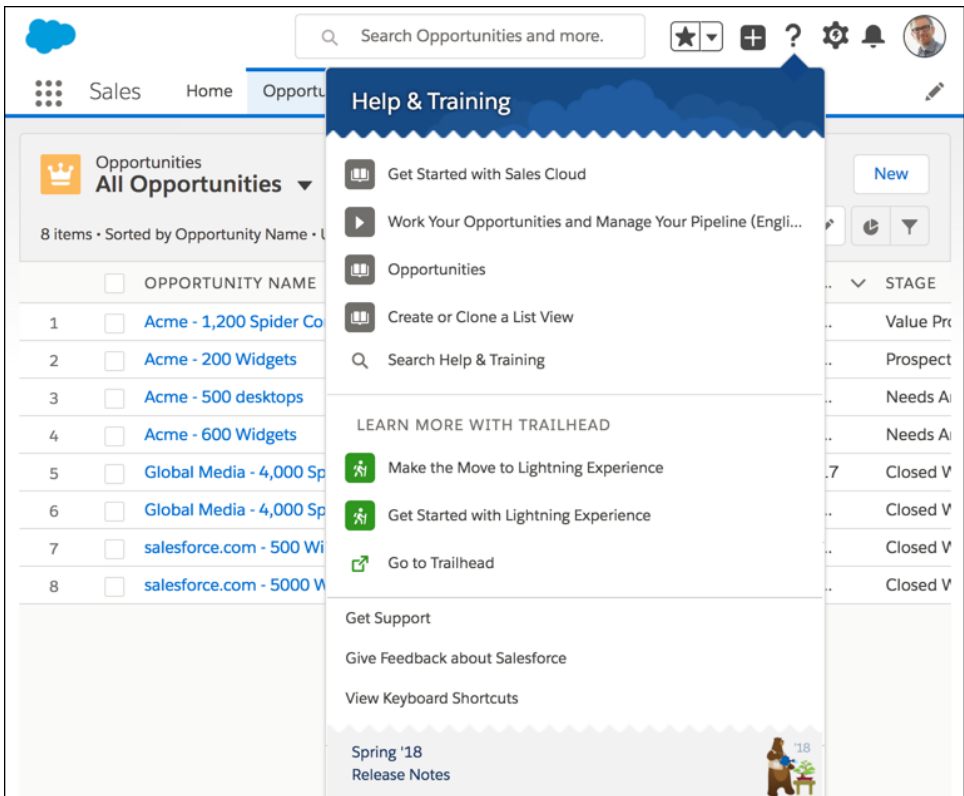
We think you'll agree you've found the perfect assistant—it has a broad perspective, yet directs you to the important things. It saves you so much time and effort that you'll probably wonder how you'd get your work done without it.

## Lightning Experience Help Menu

---

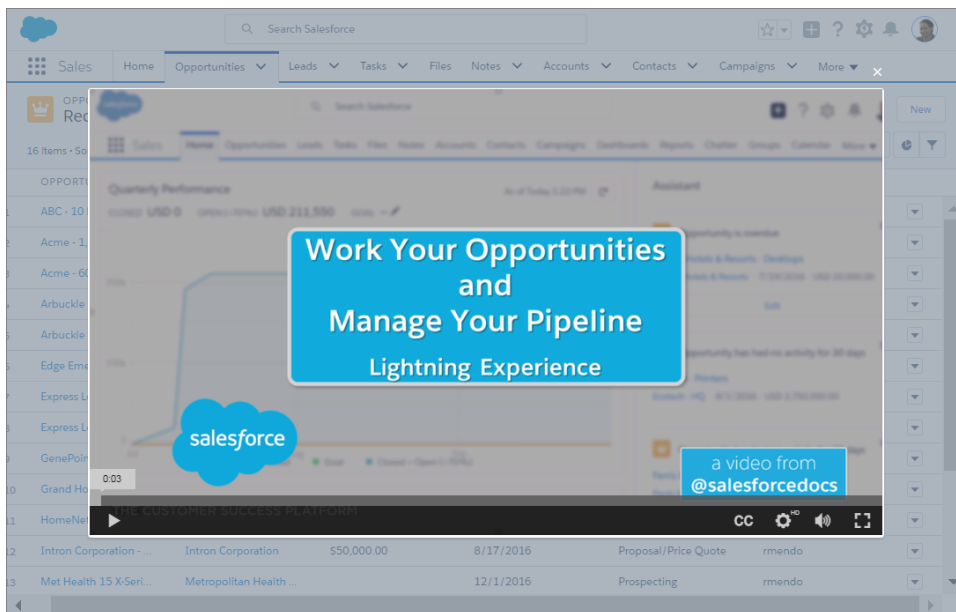
In Lightning Experience, each page has a contextual help menu with links to resources—help topics, walkthroughs, videos, developer guides, and PDFs—related to the specific tasks on that page. View help topics directly in the Help menu, and pin them to keep them open while you work.

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.



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

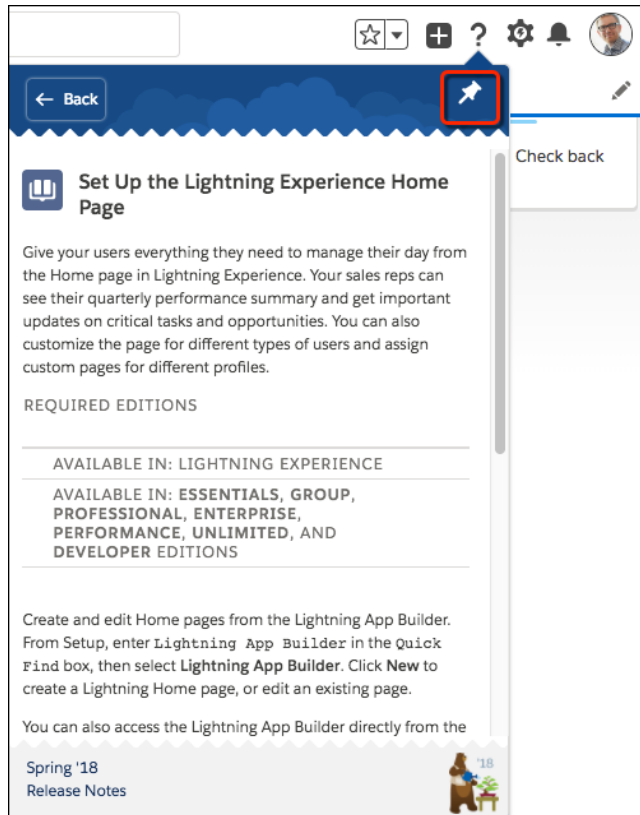
## Chapter 6 Navigation and Setup in Lightning Experience



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

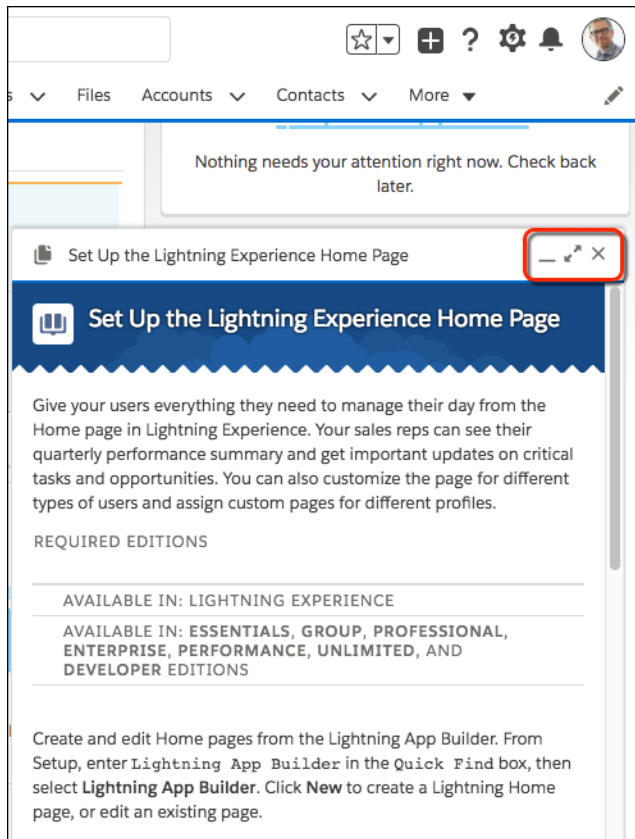
When you click a help topic (📖) in the Help menu, the topic opens in the Help menu. To keep the topic open while you work, click the pin.





The topic stays available at the bottom of your screen. You can minimize it, expand it, or close it.

## Chapter 6 Navigation and Setup in Lightning Experience




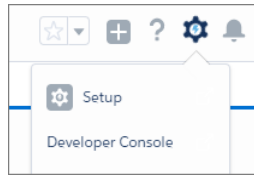
If you prefer to view the Help topic in our documentation pages, open it in a new tab using the link at the bottom of the Help menu.

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.

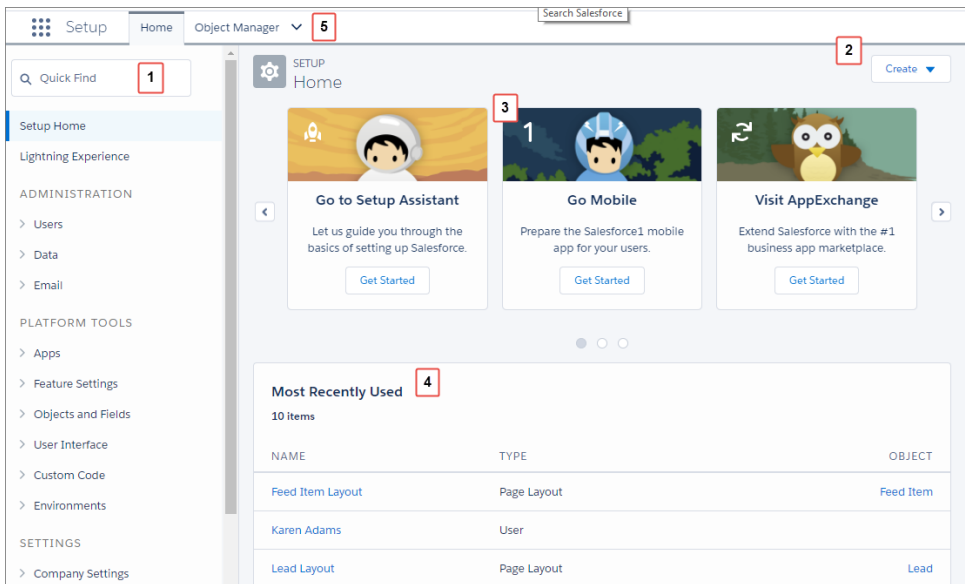
## Meet the New and Improved Setup

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  > **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.



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.

## Chapter 6 Navigation and Setup in Lightning Experience

- 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 Salesforce 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	<ul style="list-style-type: none"><li>• Your one stop for Lightning Experience customizations</li><li>• Learn best practices</li><li>• Enable/disable Lightning Experience and new features</li><li>• Setup permissions to give users access to Lightning Experience</li></ul>
Create Menu	<ul style="list-style-type: none"><li>• On every page in Setup</li><li>• Quick access to create common items</li></ul>
Object Manager	<ul style="list-style-type: none"><li>• All standard and custom objects now live in the Object Manager</li><li>• All objects now have a standard detail page that stays visible while you drill into related lists</li><li>• Infinite scroll on all objects' related lists</li></ul>
App Menu	Use it to: <ul style="list-style-type: none"><li>• Reorder apps in the App Launcher</li><li>• Make apps visible or invisible in the App Launcher</li></ul>
View Release Notes	<ul style="list-style-type: none"><li>• Links to the most recent version of the release notes</li><li>• Great point of reference for new and existing features</li></ul>

## 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.

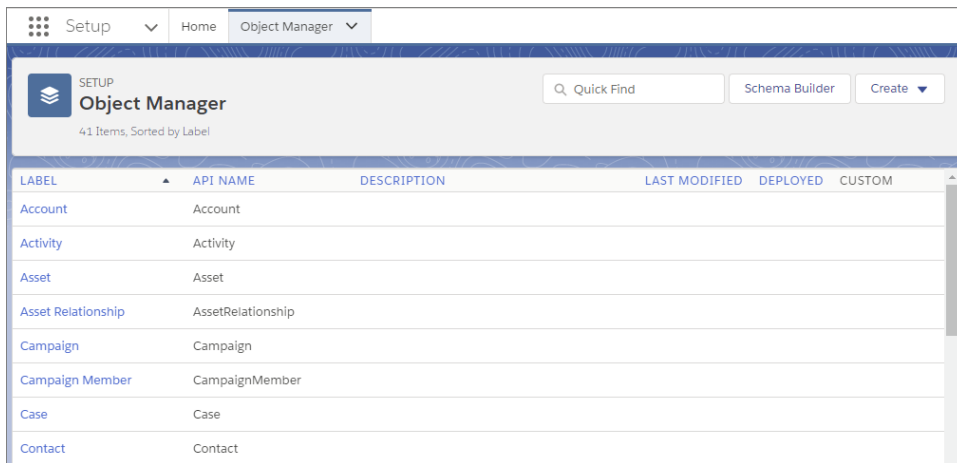
The Setup tree in Lightning Experience is limited to:

- 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

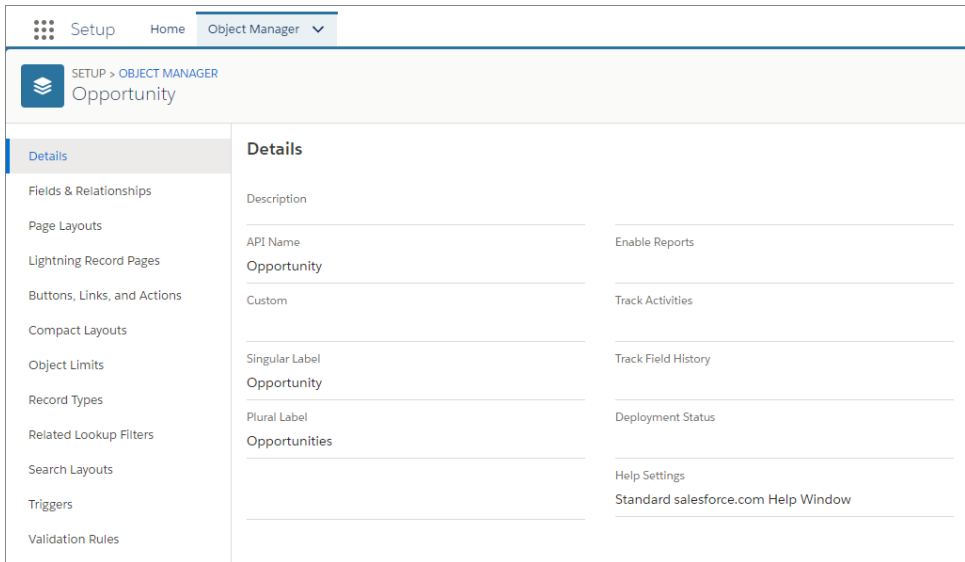
The Object Manager is a one-stop shop for managing all objects in your organization, both standard and custom. Access all objects and their related functions—fields, validation rules, page layouts, and so on—from a single entry point.



To access the Object Manager, from Setup, click **Object Manager**.

- To find an object, enter the first few characters of its label or name in the Quick Find 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. To quickly jump to a function or control, use the links in the sidebar.



## Limitations

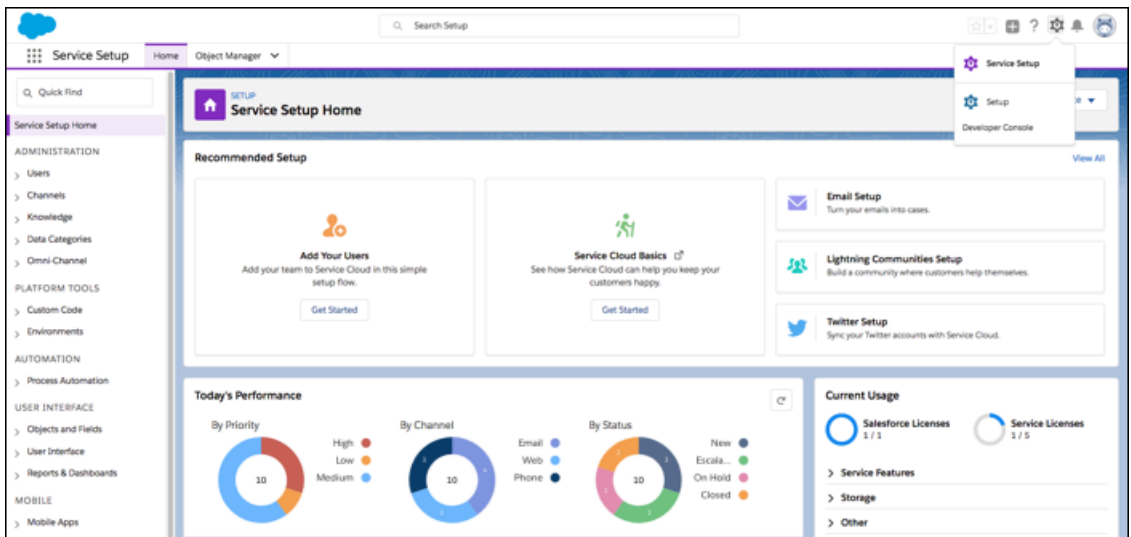
The Object Manager is limited to objects that support Lightning Experience features.

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

## Service Setup Flows

Get your service team up and running quickly on a variety of support channels. Service Setup flows offer a guided step-by-step experience to walk you through setting up various support features like email support, integrating with Twitter and Facebook, and enabling a Knowledge base. Service Setup in Lightning Experience features an intuitive setup tree and a performance metrics dashboard too.



The tiles across the top of the Service Setup Home page provide recommended setup flows, content, and one-step experiences that take the guesswork out of ramping up Service Cloud. The dashboard is customizable and shows at-a-glance metrics about your customer service team with daily performance or long-term trends. These insights help you assess and anticipate bottlenecks, and streamline calls to increase productivity.



## CHAPTER 7 Opportunities, Leads, and Selling in Lightning Experience

### In this chapter ...

- [Explore the Opportunity Workspace](#)
- [Opportunity Home](#)
- [Explore the Lead Workspace](#)
- [Accounts and Contacts in Lightning Experience](#)

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:

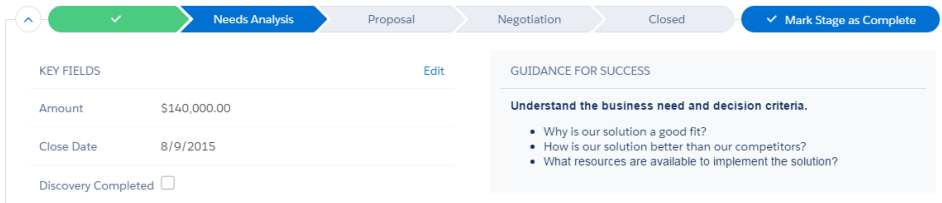
- Create and update tasks and meetings, log calls, and send email
- View key information for a deal, like its key players, account, close date, amount, owner, and stage
- Update an opportunity's stage, close date, and amount
- View or add members to an opportunity team
- View relevant, timely news about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

The screenshot displays the Salesforce Opportunity Workspace for the record 'Acme - 1,200 Widgets (Sample)'. At the top, the opportunity name is shown with a '+ Follow' button and action buttons for 'Edit', 'Delete', and 'Clone'. Below this, key fields are listed: ACCOUNT NAME (Acme (Sample)), CLOSE DATE (8/9/2015), AMOUNT (\$140,000.00), and OPPORTUNITY OWNER (Admin User). A progress bar shows the current stage as 'Needs Analysis', with previous stages 'Discovery' and 'Proposal', and future stages 'Negotiation' and 'Closed'. A 'Mark Stage as Complete' button is visible. The 'KEY FIELDS' section includes 'Amount' (\$140,000.00), 'Close Date' (8/9/2015), and a 'Discovery Completed' checkbox. A 'GUIDANCE FOR SUCCESS' section provides advice: 'Understand the business need and decision criteria.' with bullet points: 'Why is our solution a good fit?', 'How is our solution better than our competitors?', and 'What resources are available to implement the solution?'. The 'ACTIVITY' tab is active, showing a 'New Task' form with a 'Subject' field and a 'Save' button. Below the form, 'Next Steps' are listed, including 'Internal opportunity team prep call' on Aug 18 from 12:00 PM to 1:00 PM. A 'Contact Roles (3)' sidebar lists Howard Jo... (Decision Maker, Buyer), Edward Sta... (Economic Buyer, President and CEO), and Leanne To... (Influencer, VP Customer Support).

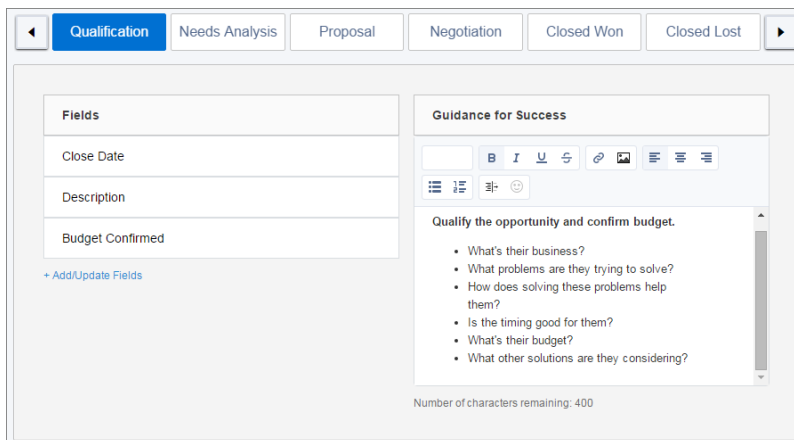
You don't have to do anything to get the opportunity workspace working for your users. However, as an administrator, you can enhance your sales users' workflow by customizing the sales path and the activity timeline to their needs.

## 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.



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.



## 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."

## Chapter 7 Opportunities, Leads, and Selling in Lightning Experience

- 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.

The screenshot displays the Activity Timeline interface, divided into two main sections: "Next Steps" and "Past Activity".

**Next Steps:**

- Internal opportunity team prep call** (Calendar icon): 10:00 AM | May 13. Start: 5/13/2016 10:00 AM, End: 5/13/2016 11:00 AM. Includes a dropdown arrow.
- Review proposal with the team** (Calendar icon): 11:30 AM | Tomorrow. Start: 5/12/2016 11:30 AM, End: 5/12/2016 12:00 PM. Includes a dropdown arrow.

**Past Activity:**

- A storm is coming...** (Envelope icon): 10:04 AM | Today. Content: "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** (Calendar icon): Apr 28. Includes a dropdown arrow.

Buttons for "More Steps" and "More Past Activity" are located at the top and bottom of the timeline, respectively.

### 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 Add Send an Email, Log a Call, New Event, and New Task Buttons to the Activity Composer.

#### **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.

**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 when an activity is expanded for events, tasks, and logged calls. Do this customization using event and task compact layouts. Even if you remove certain fields from a compact layout, they remain in the activity 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 a compact layout. 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 for events and the comments field for 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

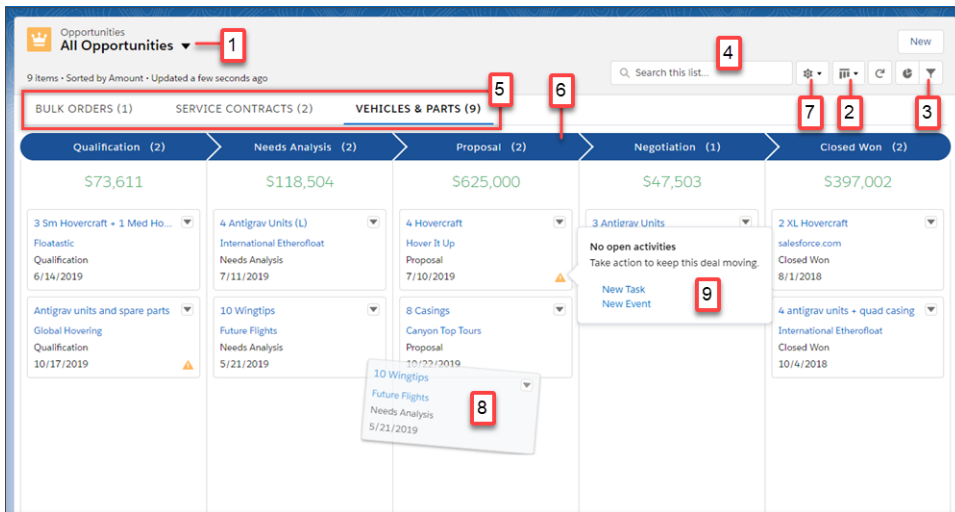
The Opportunity home page in Lightning Experience looks a lot like other object home pages at first glance. You might not notice it at first, but an awesome feature is waiting to be discovered there.

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 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.

## Chapter 7 Opportunities, Leads, and Selling in Lightning Experience



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 access a Kanban view for almost any list view.

## 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 Path. You set up paths to include specific fields and guidance for success in each stage of the lead qualification process.

The screenshot displays a Salesforce Lead record for John Steele (Sample). At the top, there is a lead card with a red star icon, the name "John Steele (Sample)", and a "+ Follow" button. To the right are "Edit", "Delete", and "Clone" buttons. Below this, the lead's details are organized into columns: COMPANY (BigLife Inc.), TITLE (Senior VP), PHONE (2) ((555) 555-1212), and EMAIL (info@salesforce.com). A progress bar shows stages: "Contacted" (active), "Nurturing", "Unqualified", and "Converted". A "Mark Status as Complete" button is on the right. The main content area is split into two sections: "KEY FIELDS" with an "Edit" link, and "GUIDANCE FOR SUCCESS". The "KEY FIELDS" section lists: Email (info@salesforce.com), Mobile ((555) 555-1212), Industry (Insurance), and No. of Employees (2,800). The "GUIDANCE FOR SUCCESS" section contains the heading "Verify contact information and qualify your lead." followed by three paragraphs of advice: "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.", and "Rehearse your call scripts. That way, you'll sound confident and the flow of your conversation will be more natural." Below this is a navigation bar with "ACTIVITY", "COLLABORATE", and "DETAILS" tabs. The "ACTIVITY" tab is active, showing a "Log a Call" button and a form with "Subject" (Call) and "Comments" fields, and a "Save" button. To the right, there is a "Learn More" section with a "Link a Twitter Account" button and a "Notes (0)" dropdown menu.

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

## Accounts and Contacts in Lightning Experience

In Lightning Experience, we optimized accounts and contacts for quick reference, so it's easy to find information and gather insights at-a-glance.

## Chapter 7 Opportunities, Leads, and Selling in Lightning Experience

The screenshot shows a Salesforce Lightning Experience contact record for Geoff Minor. The interface includes a top highlight panel (1) with fields for Title (President), Account Name (Global Media), Phone (415) 555-1212, Email (info@salesforce.com), and Contact Owner (Madison Rig...). Below this is a navigation bar with tabs for RELATED, DETAILS, and NEWS (2). The main content area is divided into three sections: a message stating 'We found no potential duplicates of this contact.' (3), a 'Related Accounts (2)' section listing Global Media and salesforce.com, and an 'Opportunities (1)' section listing 'Insightor Opp #Df4to' with details like Stage (Closed Won), Amount (\$650,000.00), and Close Date (5/10/2017). On the right side, there is an 'ACTIVITY' section with a 'CHATTER' tab, a 'Log a Call' button, and an 'Activity Timeline' showing upcoming events like 'Golf' and 'Sync with Geoff on pr...' and past activities like 'Looking forward t...' and 'Call about proposal' (4).

1. Highlight Panel: See important information right at the top of the record.
2. News and Twitter Integration: Stay informed about the latest news that affects your customers and stay connected through social media.
3. Optimized template: Easy reference to related records and at-a-glance information.
4. Activity Timeline: View emails, tasks, and events, grouped by your next steps and past activity.

There are other goodies too, like visualizing account relationships with Account Hierarchy, seeing who reports to whom with Contact Hierarchy, and relating single contacts to multiple accounts without duplicating records. Let's take a look.

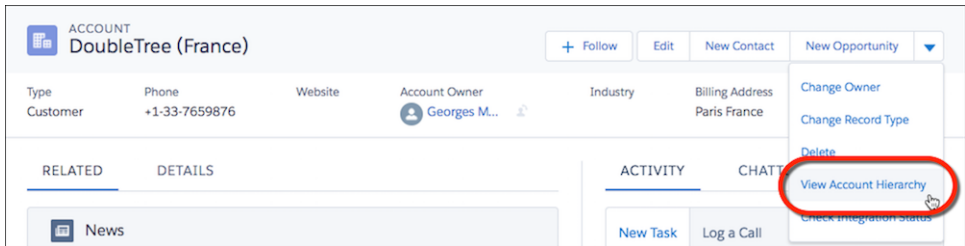
## Account Hierarchy

Same as in Salesforce Classic, Account Hierarchy gives sales reps a bird's-eye view of the relationships between parent accounts and their subsidiary accounts. But there's a bonus when using this feature in Lightning Experience—you can customize the hierarchy columns to show the information that's most useful to your sales reps.


The Parent Account field connects an account to its parent. In Lightning Experience, display an account's hierarchy from the Actions dropdown menu on the account record page.




## Chapter 7 Opportunities, Leads, and Selling in Lightning Experience



The screenshot shows the Salesforce account page for 'DoubleTree (France)'. The page includes fields for Type (Customer), Phone (+1-33-7659876), Website, Account Owner (Georges M...), Industry, and Billing Address (Paris France). Below these fields are tabs for 'RELATED' and 'DETAILS'. The 'RELATED' tab is active, showing a 'News' section. On the right side, there is an 'ACTIVITY' section with 'New Task' and 'Log a Call' buttons. A dropdown menu is open, showing options: 'Change Owner', 'Change Record Type', 'Delete', 'View Account Hierarchy' (highlighted with a red circle), and 'Check Integration Status'.

 **Note:** If you customized the Salesforce Mobile and Lightning Experiences Actions section on the account page layout before Spring '17, you need to add the View Account Hierarchy action to the page layout to make it available to your users.

Users expand or collapse parts of a hierarchy as they navigate through it. They can view up to 2,000 accounts from each point where they enter a hierarchy.

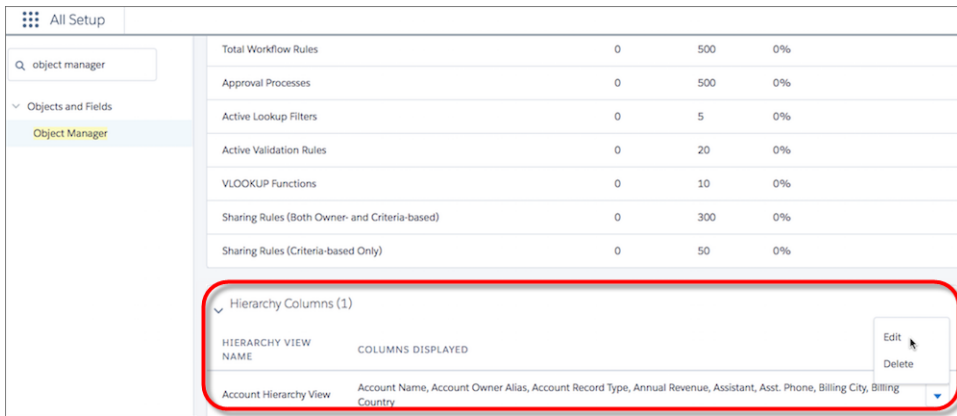


The screenshot shows the 'Account Hierarchy' table for the 'DoubleTree (France)' account. The table has four columns: 'ACCOUNT NAME', 'ANNUAL REVENUE', 'BIRTHDATE', and 'FIRST NAME'. The rows are as follows:

ACCOUNT NAME	ANNUAL REVENUE	BIRTHDATE	FIRST NAME
▼ Hilton Inc	\$125,876,987		
> Canopy	\$1,456,543		
> Conrad	\$4,765,789		
> Curio	\$4,766,123		
▼ DoubleTree	\$5,672,341		
> DoubleTree (Canada)	\$987,654		
> <b>CURRENT</b> DoubleTree (France)	\$109,876		
> DoubleTree (Germany)	\$564,123		
> DoubleTree (Italy)	\$1,765,432		

By default, account hierarchies display the same columns as the standard Recently Viewed Accounts list view. But you can customize the hierarchy columns independently of the list view so sales reps see the information that's most useful to them when they work in Lightning Experience. From Setup in Lightning Experience, click the **Object Manager** tab. In Account, click **Hierarchy Columns**, and then edit the columns.

## Chapter 7 Opportunities, Leads, and Selling in Lightning Experience



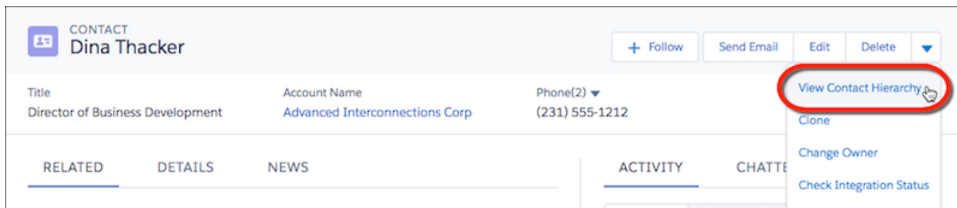
object manager				
Total Workflow Rules	0	500	0%	
Approval Processes	0	500	0%	
Active Lookup Filters	0	5	0%	
Active Validation Rules	0	20	0%	
VLOOKUP Functions	0	10	0%	
Sharing Rules (Both Owner- and Criteria-based)	0	300	0%	
Sharing Rules (Criteria-based Only)	0	50	0%	

Hierarchy Columns (1)		Edit	Delete
HIERARCHY VIEW NAME	COLUMNS DISPLAYED		
Account Hierarchy View	Account Name, Account Owner Alias, Account Record Type, Annual Revenue, Assistant, Asst. Phone, Billing City, Billing Country		

## Contact Hierarchy

Lightning Experience also includes Contact Hierarchy, which allows reps to generate an org chart of who reports to whom in an account. A hierarchy is derived from information in the Reports To field on contact records. In Lightning Experience, view the contact hierarchy from the Actions dropdown menu on the contact record page.



CONTACT  
Dina Thacker

+ Follow Send Email Edit Delete

Title: Director of Business Development  
Account Name: Advanced Interconnections Corp  
Phone(2): (231) 555-1212

View Contact Hierarchy  
Clone  
Change Owner  
Check Integration Status

CONTACT > DINA THACKER		
Contact Hierarchy		
NAME	TITLE	PHONE
▼ Kathy Fitzpatrick	President	(231) 555-1212
Brenda McClure	Chief Financial Officer	(231) 555-1212
▼ Mark Metz	Chief Executive Officer	(231) 555-0100
Dina Thacker <b>CURRENT</b>	Director of Business Development	(692) 555-1212
> Sean Murphy	Director of Marketing	(231) 555-0100
> Steve McDonald	Director of Information Technology	(231) 555-1212

Hovering over a contact's name shows more details.

As with Account Hierarchy, you can customize the contact hierarchy columns so you can show the details that are most useful to your sales team when they work in Lightning Experience. From Setup in Lightning Experience, click the **Object Manager** tab. In Contact, click **Hierarchy Columns**, and then edit the columns.

## Contacts to Multiple Accounts

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.

## Chapter 7 Opportunities, Leads, and Selling in Lightning Experience

When the feature is set up, account records include the Related Contacts related list and contact records include the Related Accounts related list.

The screenshot displays a Salesforce Account record for 'salesforce.com'. The header includes fields for Type (Customer), Phone (415) 901-7000, Website (http://www.salesforce.com), Account Owner (Madison Riggs), Industry (Technology), and Billing Address (The Landmark @ One Market, Suite 300, San Francisco, CA 94105 USA). Below the header, there are tabs for 'RELATED' and 'DETAILS'. The 'RELATED' section is divided into 'News' and 'Related Contacts (6+)'. The 'News' section shows four news items from Salesforce.com, Yahoo Finance, and Google.com. The 'Related Contacts' section lists four contacts: Abraham Adam (Software Development Intern), Atfal Ahmed (Senior Payroll Specialist), Alexa Adams (Nonprofit Account Coordinator), and Ansar Ahmed (Manager Sales Engineering). To the right, there are tabs for 'ACTIVITY' and 'CHATTER'. The 'ACTIVITY' section shows a list of 'Next Steps' and 'Past Activity'. The 'Next Steps' include 'Review proposals' (due Jul 7) and 'Call Jono' (due Jun 29). The 'Past Activity' includes 'event 1' (3:00 PM Mar 30) and 'schedule meeting' (Mar 29).

## CHAPTER 8 Service Cloud in Lightning Experience

### In this chapter ...

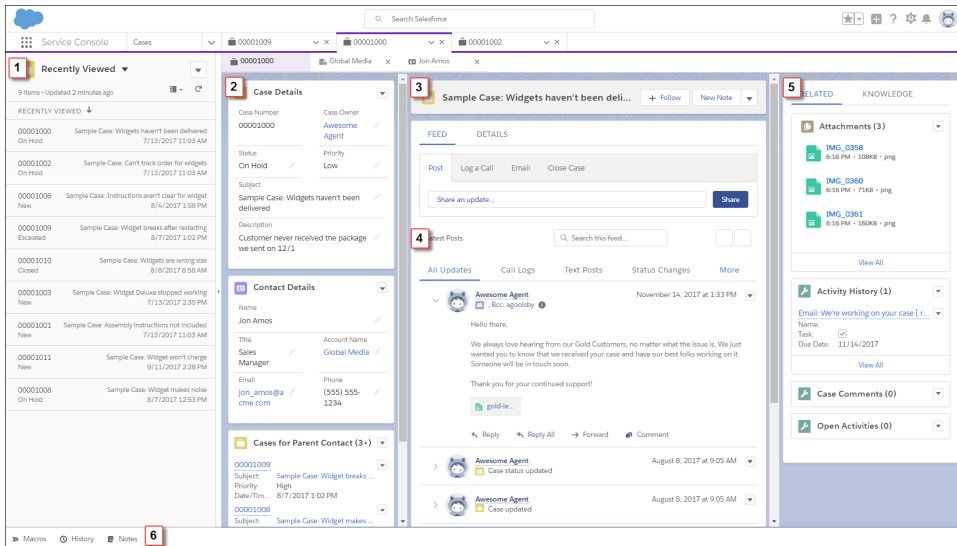
- The Fast-As-Lightning Service Console
- Be a Knowledge Sponge
- Snap-ins Chat for Fast, Snappy Service
- Analyze the Service Analytics App
- Get Out There with Field Service Lightning
- Omni-Channel Knows All
- Social Customer Service

Wondering which Service Cloud features are available in Lightning Experience? Well, wonder no longer! Let's talk Service Cloud.

# The Fast-As-Lightning Service Console

The Lightning Service Console is a standard Salesforce Lightning console app that's designed to meet all your service team's needs. Plus you can customize the console to make it your own.

Here's what you get in the standard Service Console app.



Here's how the standard Service Console app helps you out.

## Manage cases faster

Split view shows a list view at the same time as workspace tabs and subtabs, letting your support agents manage multiple cases on a single screen (1).


## See it all at once

A preconfigured, three-column layout puts all the important information on the same page, and minimizes clicks and scrolling.

- In the first column, case details, contact details, and related cases for the parent account are displayed using Related Record and Related List Single components (2).
- The second column includes a highlights panel that's front and center (3). The compact case feed lets you see more of your case updates (4).
- The third column includes related lists and Knowledge articles are at the ready (5).

### Keep up on all the details

The preconfigured utility bar provides fast access to History and Notes (6). You can also customize the utility bar to include other tools like Macros, Omni-Channel, and the Open CTI Softphone.


 **Note:** For users to see and use the Notes utility, Notes must be enabled in your org. For users to see and use articles in the Knowledge component, Lightning Knowledge must be enabled in your org.

## Best Practice Makes Perfect

As awesome as the default Lightning Service Console is, with a little polish, you can tailor it to meet your specific service needs.

We recommend that you:

- Update the name, description, and branding for the app
- Add other items, such as quick text, macros, products, and contracts (the Service Console app includes cases, contacts, accounts, reports, dashboards, chatter, and Home by default)
- Add other utilities, such as an Open CTI softphone, Omni-Channel, Macros, and Community 360
- Assign the app to the appropriate user profiles, like your service agent profiles

 **Note:** The Service Console app is automatically assigned to all standard and custom user profiles. You can modify these assignments to make sure that only the right people have access to the app.

The only thing you can't change is the app's Developer Name.

## Be a Knowledge Sponge

---

So what's so great about Lightning Knowledge? Everything you need is all in one place! It provides a high-powered yet streamlined way to manage your knowledge base. You get the benefits of standard objects that work just like other objects in Salesforce. Lightning Knowledge has changed the way Knowledge works in Salesforce. Standard record types replace article types, and the Knowledge component for Lightning Service Console replaces Knowledge One for the Service Console in Salesforce Classic. These changes make it much easier for you to set things up.

Your support agents can search, view, author, and manage articles from a single Knowledge home page in Lightning Experience. Plus, they can do the following authoring actions without leaving Knowledge home.

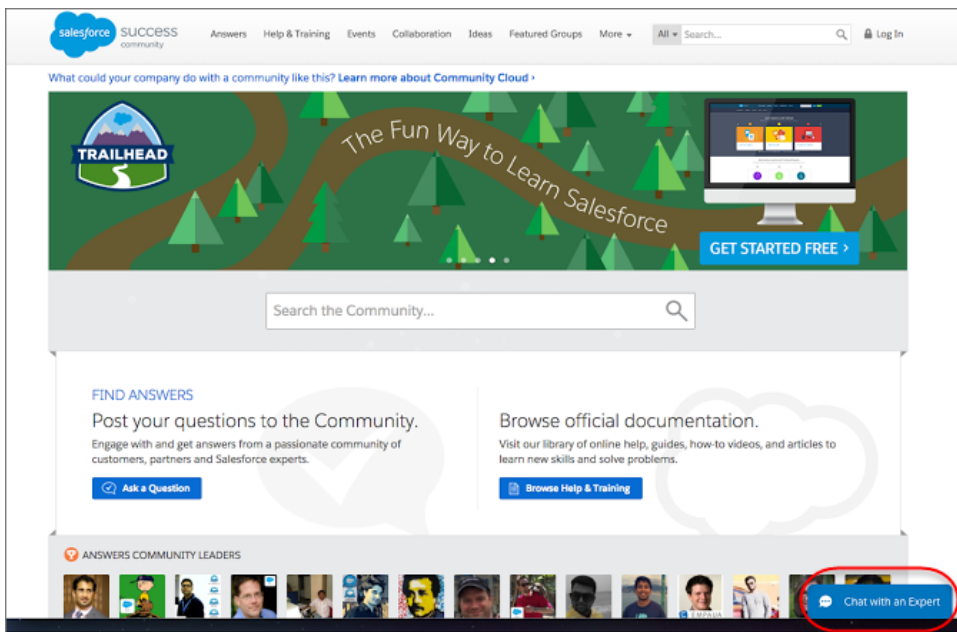
## Chapter 8 Service Cloud in Lightning Experience

Published Articles	Article Drafts	Archived Articles
Create a new article	Create a new article	Create a new article
Archive a published article	Publish a draft article	Restore an archived article
Edit a published article as a new draft	Delete a draft article	
	Edit a draft article	

## Snap-ins Chat for Fast, Snappy Service

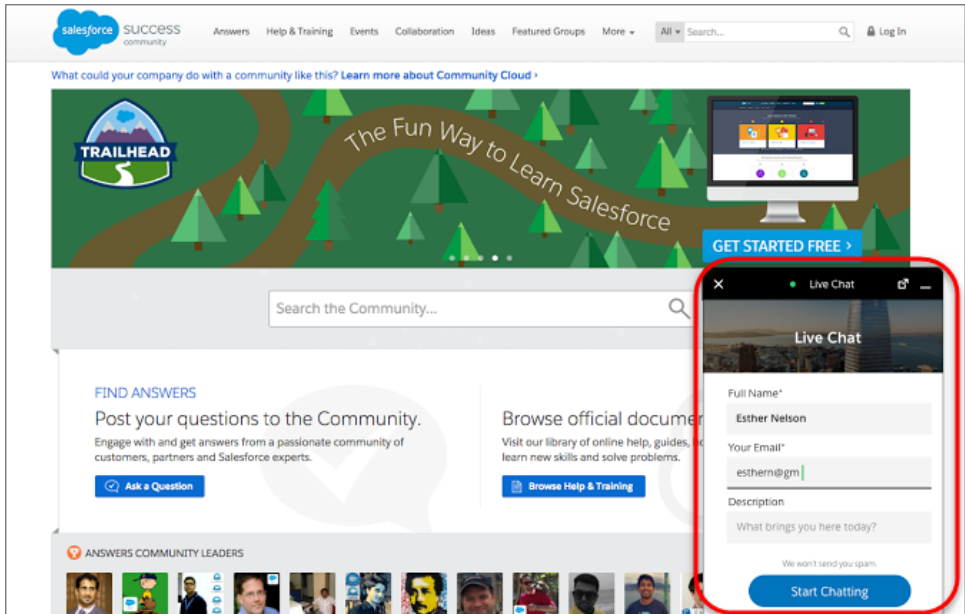
Snap-ins chat allows customers to get quick answers to their questions by chatting with an agent while browsing your site. So, it's a great idea to add the Snap-ins chat widget to your website! Snap-ins chat uses Live Agent, but in Lightning Experience you have a simpler setup on the Service Console.

What the customer sees: the chat widget button sits on your web page, and when someone has a question, they simply click the button to launch the chat.



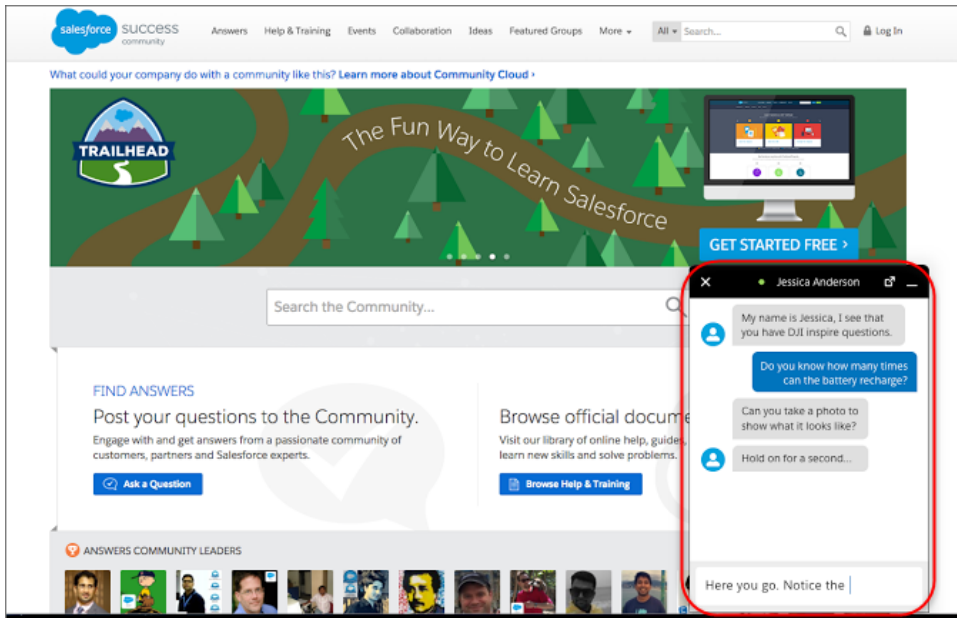


Customers fill out the brief pre-chat form, which helps agents gather basic information about the customer, like their contact information and their needs.

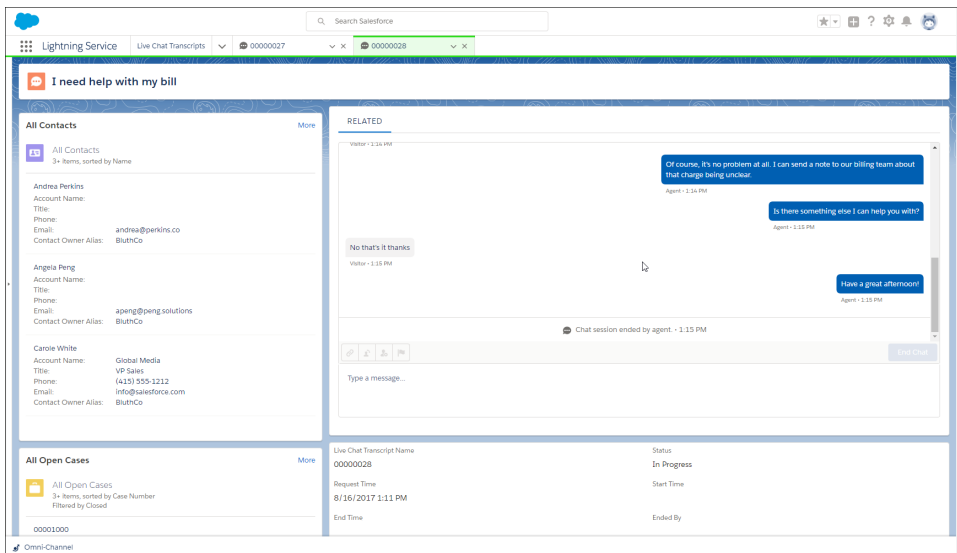


Customers can chat while viewing your web page, and they can minimize the chat window whenever it's in their way. The chat widget persists across your web pages so customers can keep browsing while chatting with an agent.

## Chapter 8 Service Cloud in Lightning Experience



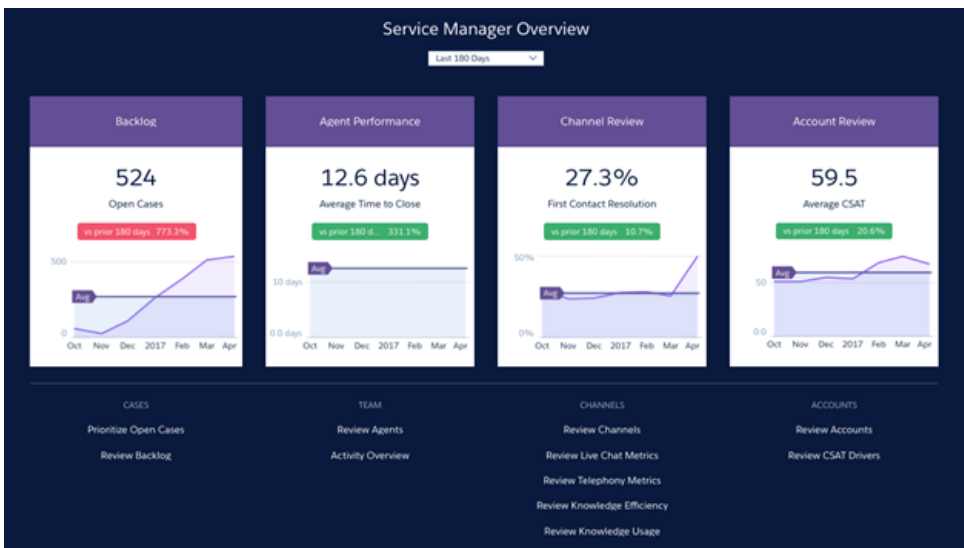
This example shows what support agents see when they chat with customers via Snap-ins Chat. Snap-ins Chat sits directly in the Service Console, allowing a quick and easy response to customers.



## Analyze the Service Analytics App

Help your support team use data to uncover key insights and drive the success of your service business. The Service Analytics app shares best-practice key performance indicators (KPIs) about your service data in a single place. Even better, the app tailors this information by role, so everyone on the team gets the right information at the right time.

Using the prebuilt dashboards in the Service Analytics app, service managers can quickly view average case closing times, customer satisfaction, and trending, historical, and peer benchmarks. They can also get insight into the team’s use of knowledge articles to resolve cases, and other data so they can quickly take appropriate actions.



Another set of dashboards—called sidebars—are specifically for support agents. For a given case, an agent can view customer history, number of cases, and CSAT. Ensure that agents have fast access to this information by embedding these sidebars on key Salesforce pages, such as the service console.

Sounds complicated and time-consuming to set up? Most of the hard work is already done for you! We built the complex queries, formulas, and ratios that draw from your service and sales data, and assembled them into easy-to-read visualizations. All you do is run a built-in configurator and answer a few questions about the data and fields you’d like to see. Woohoo!

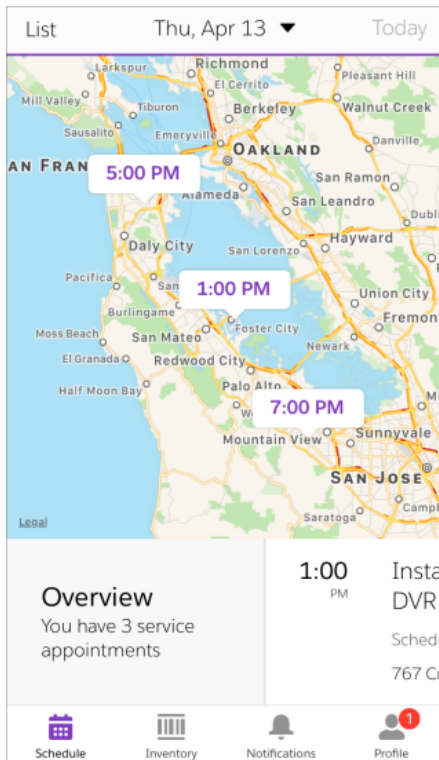
# Get Out There with Field Service Lightning

Field Service Lightning is a mobile-friendly field service hub for your service team. Your agents can easily connect with people who work at customer sites, or who work on the go in delivery vehicles. Field Service Lightning helps everyone on the team make smarter decisions with visibility into operational performance and key performance indicators.

Running a field service business means managing many moving parts. With Field Service Lightning, your agents, dispatchers, and field technicians get the tools to manage work orders, service resources, and scheduling. But what about specifics? How about:

- Create service resource records that represent your field service technicians and add details about their skills, service territories, and availability.
- Set up multilevel service territories that represent the regions where your technicians work.
- Track your product inventory and service vehicle locations.
- Schedule one-time or recurring appointments for customers and add details about technician preference and required skills.
- Standardize your business’s field service tasks with maintenance plans and templates.
- Keep customers informed about service progress with on-site service reports.

The screenshot displays the Field Service Lightning interface. At the top, there are navigation tabs for 'Service Appointments', 'Service Resources', 'Resource Absences', and 'Accounts'. The user is logged in as 'YD V-One - yd-fsl@one.demo'. The main view is a Gantt chart for the period 'Thurs, April 13 2017 - Friday, April 14 2017'. The chart shows service appointments for various technicians across different service territories. On the left, there is a 'Todo' list with a search bar and a table of appointments. The table lists appointments with IDs like SA-0410, SA-0804, SA-0808, SA-0815, and SA-0816, all with a status of 'None' and various start times. The Gantt chart itself shows colored bars representing appointments for technicians like Lamar Smith, Marino Castanon, Pam Lawison, Paul Morrison, Thomas Taylor, Janice Gonzales, Louise Martin, and Paul Morrison. A blue notification box indicates that Paul Morrison has been 'Relocated to San Francisco'.



## Omni-Channel Knows All

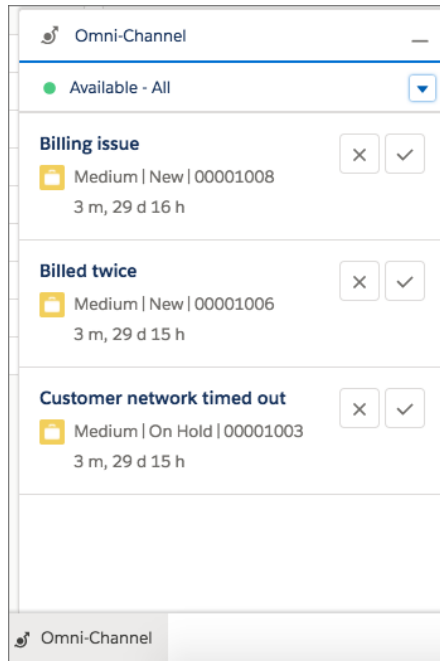
Omni-Channel helps your service center route any type of incoming work item—including cases, chats, or leads—to your most qualified, available agents. It’s flexible and customizable, and you can configure it without writing code. Pretty rad!

Use Omni-Channel to manage the priority of work items for agents and balance the distribution of assignments. You can also define which agents work on different types of assignments, such as leads or sales inquiries, and other assignments that help with support questions.

Omni-Channel routes all these assignments to the correct agents automatically, which your agents will love. They no longer have to pick and choose work assignments from a queue, which saves everyone time, effort, and brainpower.

## Chapter 8 Service Cloud in Lightning Experience

In Lightning Experience, the Omni-Channel utility delivers work right to your Lightning Service Console users via requests routed through Omni-Channel. Console users can set their presence status and accept or decline work depending on your Omni-Channel settings.

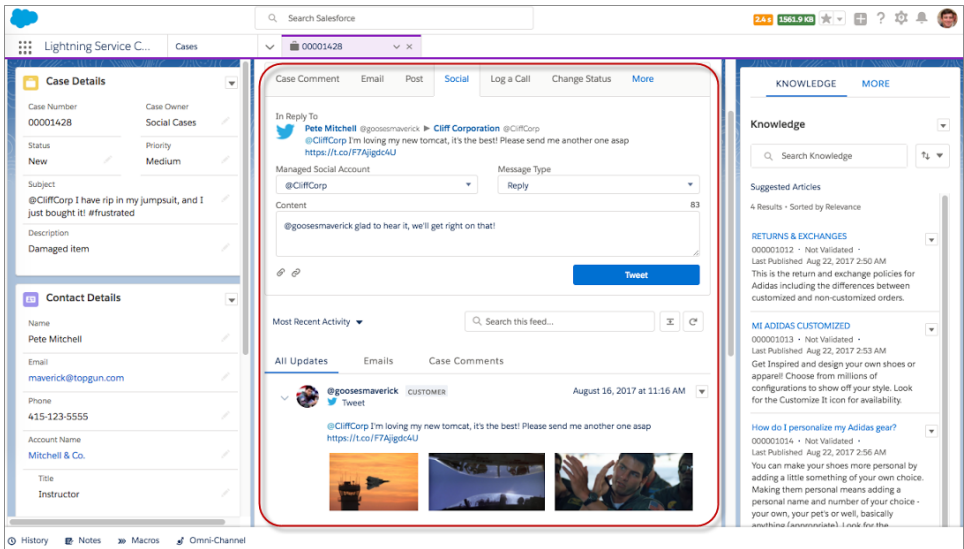


## Social Customer Service

---

Help your support team deliver personalized customer service from one of your most important channels—social media. Social Customer Service integrates with Twitter, Facebook, Instagram, and YouTube to deliver customer conversations directly to your agents. Setup is easier than ever in Lightning Experience, including Social Business Rules to quickly define how social cases are handled.

Support agents get the tools to manage cases originating from social media and engage customers by responding directly in social networks.



Inbound and outbound social posts appear as items in the case feed, making it easy to follow conversations. Attachments to posts are also viewable right in the feed. Agents can respond to posts using the Social action in the publisher and include an attachment or a Direct Message prompt. They can also like conversations, reply privately to customers, hide comments on Facebook, and delete conversations managed by your company's social accounts.





## CHAPTER 9 Explore More of Lightning Experience

### In this chapter ...

- Custom Themes and Branding
- The Home Page
- Object Home Pages and List Views
- How Other Objects are Organized in Lightning Experience
- Chatter in Lightning Experience
- File Management and Collaboration
- Tasks, Calendar, and Events

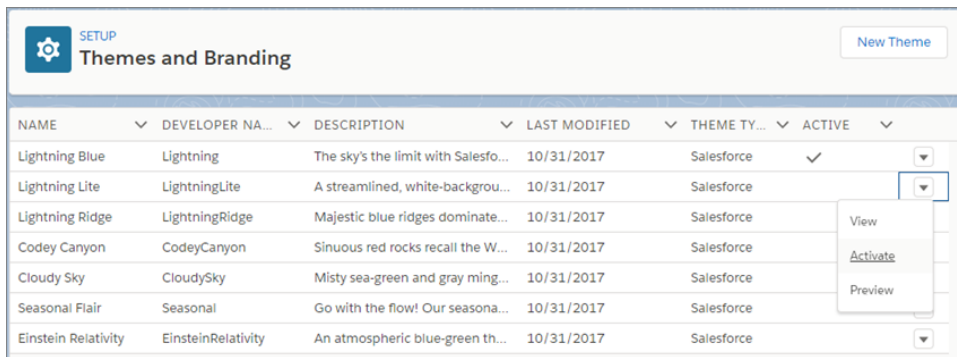
Opportunities and leads aren't the only places we've made improvements. There are slick new features to discover elsewhere, like custom brand images and colors in your org, the new Home page, revamped object home pages and enhanced list views, and enhanced Chatter and activities options.

## Custom Themes and Branding

Rally your team around your brand and important initiatives, like a new product launch, with a set of custom brand images and colors in your org. We call them *themes*. You can choose one of Salesforce's built-in themes or create your own custom themes with just a few clicks.

Theme and brand-based colors show up in various places in the user interface, including page backgrounds, the global header, navigation bar, tabs, and buttons. You can also use a theme's brand image on the Lightning Experience loading page. By default, orgs use the built-in Lightning Blue theme.

From Setup, enter *Themes and Branding* in the Quick Find box, then select **Themes and Branding**. You can view, preview, and activate an existing theme or click **New Theme** to create your own.



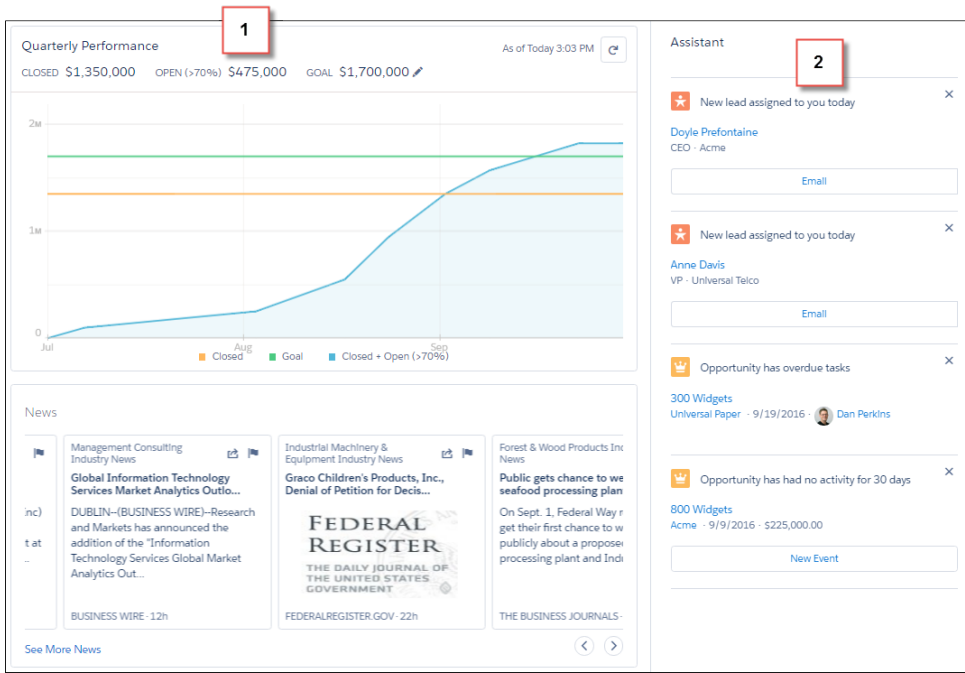
NAME	DEVELOPER NA...	DESCRIPTION	LAST MODIFIED	THEME TY...	ACTIVE
Lightning Blue	Lightning	The sky's the limit with Salesfo...	10/31/2017	Salesforce	✓
Lightning Lite	LightningLite	A streamlined, white-backgrou...	10/31/2017	Salesforce	
Lightning Ridge	LightningRidge	Majestic blue ridges dominate...	10/31/2017	Salesforce	
Codey Canyon	CodeyCanyon	Sinuous red rocks recall the W...	10/31/2017	Salesforce	
Cloudy Sky	CloudySky	Misty sea-green and gray ming...	10/31/2017	Salesforce	
Seasonal Flair	Seasonal	Go with the flow! Our seasona...	10/31/2017	Salesforce	
Einstein Relativity	EinsteinRelativity	An atmospheric blue-green th...	10/31/2017	Salesforce	

Upload a brand image, and choose brand colors. Custom themes modify the brand color to make it accessible. However, you can override the accessible brand color. Optionally, upload a page background image, default banners, and avatars for groups and user profiles.

## 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.



## 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
- Overdue opportunities

Opportunities are overdue if they're still open after the Close Date. These updates stop appearing if it's been over 8 days since the Close Date.








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 by default. You can customize the page to add a feed, or access Chatter 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:

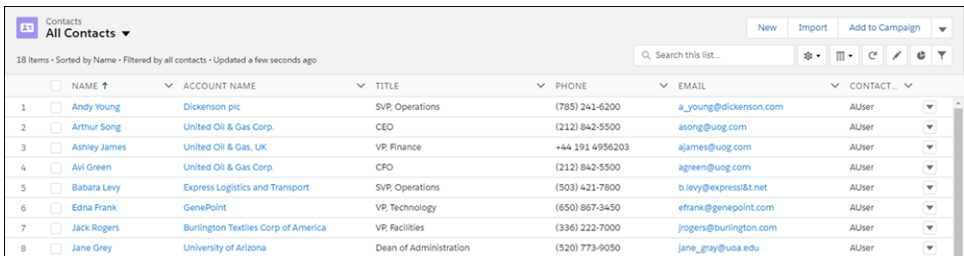
**Table 3: Required Permissions for Home Features**

Permission or Setting	Performance Chart	Key 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			
"Run Reports" user permission enabled for users			

Permission or Setting	Performance Chart	Key Deals	Assistant
Closed opportunities or open opportunities with a probability over 70% during the current fiscal quarter			
Read access to the Lead object			

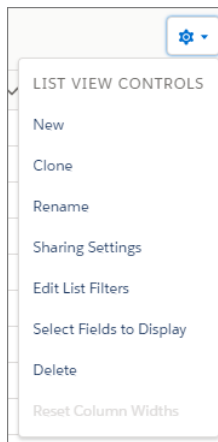
## Object Home Pages and List Views

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



	NAME ↑	ACCOUNT NAME	TITLE	PHONE	EMAIL	CONTACT...
1	<input type="checkbox"/> Andy Young	Dickenson plc	SVP, Operations	(785) 241-6200	a_young@dickenson.com	ALUser
2	<input type="checkbox"/> Arthur Song	United Oil & Gas Corp.	CEO	(212) 842-5500	asong@uog.com	ALUser
3	<input type="checkbox"/> Ashley James	United Oil & Gas, UK	VP, Finance	+44 191 4956203	ajames@uog.com	ALUser
4	<input type="checkbox"/> Avi Green	United Oil & Gas Corp.	CFO	(212) 842-5500	agreen@uog.com	ALUser
5	<input type="checkbox"/> Babara Levy	Express Logistics and Transport	SVP, Operations	(503) 421-7800	b.levy@expresslst.net	ALUser
6	<input type="checkbox"/> Edna Frank	GenePoint	VP, Technology	(650) 867-3450	efrank@genepoint.com	ALUser
7	<input type="checkbox"/> Jack Rogers	Burlington Textiles Corp of America	VP, Facilities	(336) 222-7000	jrogers@burlington.com	ALUser
8	<input type="checkbox"/> Jane Grey	University of Arizona	Dean of Administration	(520) 773-9050	jane_gray@uoa.edu	ALUser

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



## Chapter 9 Explore More of Lightning Experience

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.

## Considerations When Working with Object Home Pages and List Views

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 dropdown 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 dropdown 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 dropdown 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 view items.

## How Other Objects are Organized in Lightning Experience

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.

The screenshot shows a Salesforce Lightning Experience record page for a Contact named Mr. Marc Benioff. The page is organized into several sections:

- Header:** Includes the contact's name, profile picture, and action buttons: Follow, Edit, New Case, and New Note.
- Details:** A horizontal bar containing key information: Title (Executive Officer), Account Name (salesforce.com), Phone (415) 901-7000, Email (Info@salesforce.com), and Contact Owner (Albert Einstein).
- Main Content Area:**
  - RELATED TAB:** Contains a message: "We found no potential duplicates of this contact." Below it is a "Related Accounts (2)" section showing a relationship between salesforce.com and UCSF Benioff Children's Hospital, with roles like Direct and Executive Sponsor. Below that is an "Opportunities (1)" section showing a record for "5,000 Astro Dolts" with details like Stage (Negotiation/Review), Amount (\$1,000.00), and Close Date (1/31/2018).
  - DETAILS TAB:** Currently selected, showing the main content area.
  - NEWS TAB:** Currently inactive.
- Right Hand Column:**
  - ACTIVITY TAB:** Currently selected, showing options for New Task, New Event, Email, and Log a Call, along with a "Create a task..." input and an "Add" button.
  - CHATTER TAB:** Currently inactive.
  - Activity Timeline:** Shows "No next steps. To get things moving, add a task or set up a meeting." and "Past Activity" with a recent event: "meeting to show protot... 4:00 PM | Jan 3".

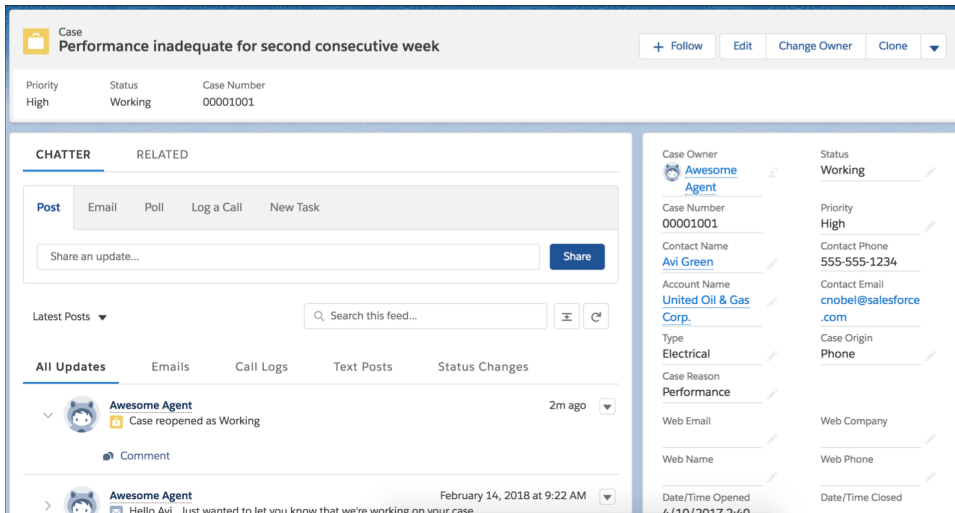
The highlights panel is in the same place across all objects. But here there's no Path, and the Activity and Chatter tabs are together in the right hand column. Related lists are in their own tab in the main part of the page, alongside the Details tab.

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.

## Cases

Cases look slightly different than other records. Cases are feed-first and display a Chatter tab first, rather than record details or related information. The case feed helps support agents collaborate and work with cases faster. Details appear next to the feed. You don't see the Activity tab because this information displays directly in the Chatter case feed. In the Service Console app, cases display in a three-column page layout, with the Chatter publisher in the Feed tab.

## Chapter 9 Explore More of Lightning Experience



To fully benefit from the feed-first design on cases, recreate your Salesforce Classic case feed publisher actions in Lightning Experience. One benefit of recreating the publishers actions is that they appear on mobile devices, whereas the Salesforce Classic 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

Edit Create Fund Request Submit Forecast

RELATED DETAILS

**System Information**

Expense Report #  
EXP-20070803-0000

Owner  
Admin User [Change Owner](#)

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

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

**Information**

Employee Name  
Ely East

Status  
Submitted

Billable

Manager For Approval  
Kasey Lee

Description

Total  
\$900.00

Week Ending  
3/30/2015

Department #

**ACTIVITY**

New Task Create e... Log a Call

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 Profiles

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.

The screenshot displays a Salesforce user profile for Pat Martin. At the top, there is a header image of a landscape with a river and mountains, and three buttons: "New Note", "User Detail", and "Edit". Below the header is the user's profile picture, name "Pat Martin", and title "Sales Manager", along with a "+ Follow" button. The profile is divided into two main sections: "About" and "Contact".

**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		

On the right side of the profile, there are four summary boxes:

- Files (0)
- Groups (0)
- Followers (0)
- Following (0)

## Chatter in Lightning Experience

Wondering what's up with Chatter in Lightning Experience? Draft posts and more accessible input focus in the publisher. More, better Chatter!

## No More Lost Posts from Session Timeouts

Have you ever had that feeling where you lost a brilliant or lengthy post because of a session timeout? No more! Admins can flip one switch in Setup and give you Chatter draft posts.

**SETUP**  
**Chatter Settings**

## Chatter Settings

Chatter is a corporate network that lets your users work together, talk to each other, and share information, all in real time.

[Edit](#)

### Chatter Settings

Turn on Chatter and Global Search features. We have given you a head start—your users may auto-follow a few people records by default and your search box is in the header. [Learn More...](#)

Enable

### Groups

Modify Chatter group settings.

- Allow Group Archiving  [i](#)
- Allow Records in Groups  [i](#)
- Enable Unlisted Groups  [i](#)

### Rich Link Previews in Feed

Show rich content in the feed. Convert links in posts into embedded videos, images, and article previews. Rich content is provided by Embed.ly, a third-party service. Previews are only available for links to supported sites. We don't share any private content with Embed.ly, just the URL.

Allow Rich Link Previews

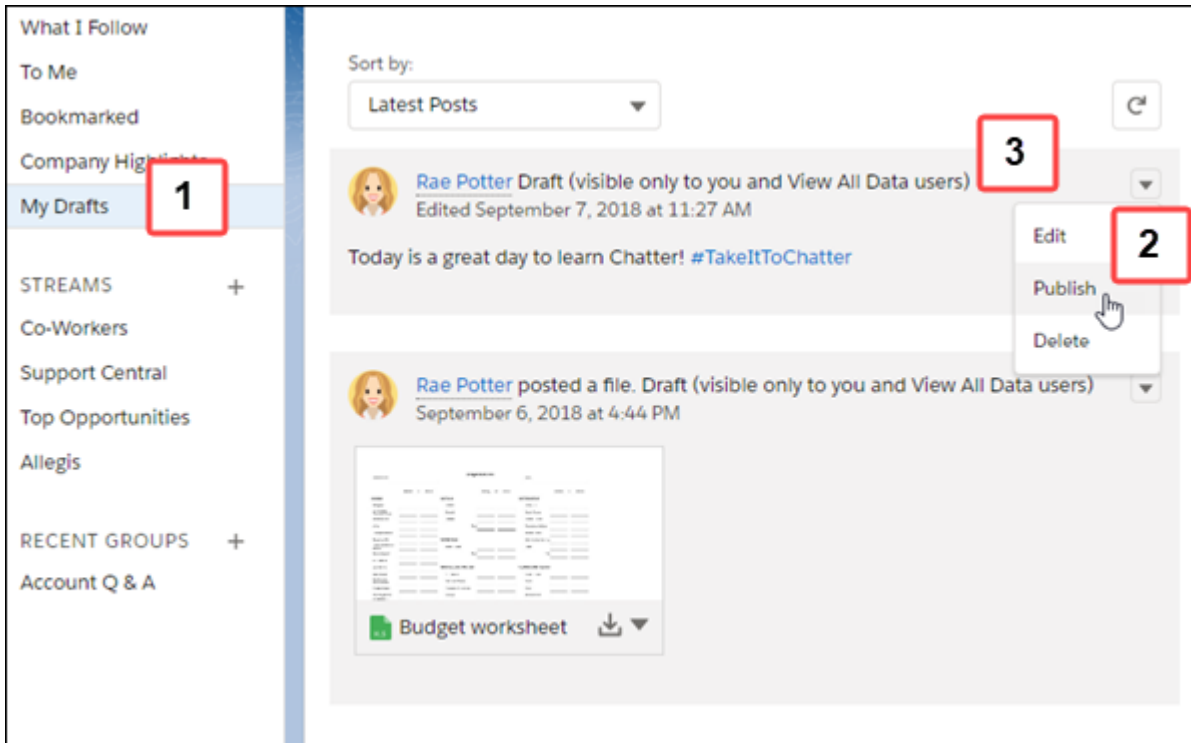
### Draft Posts

Turn on Draft Posts and the My Drafts feed.

Allow draft posts

After your admin selects Allow draft posts, your drafts are saved every seven seconds. A new My Drafts feed (1) appears on the Chatter tab. Go to My Drafts to view, edit, publish, or delete a draft (2).

## Chapter 9 Explore More of Lightning Experience

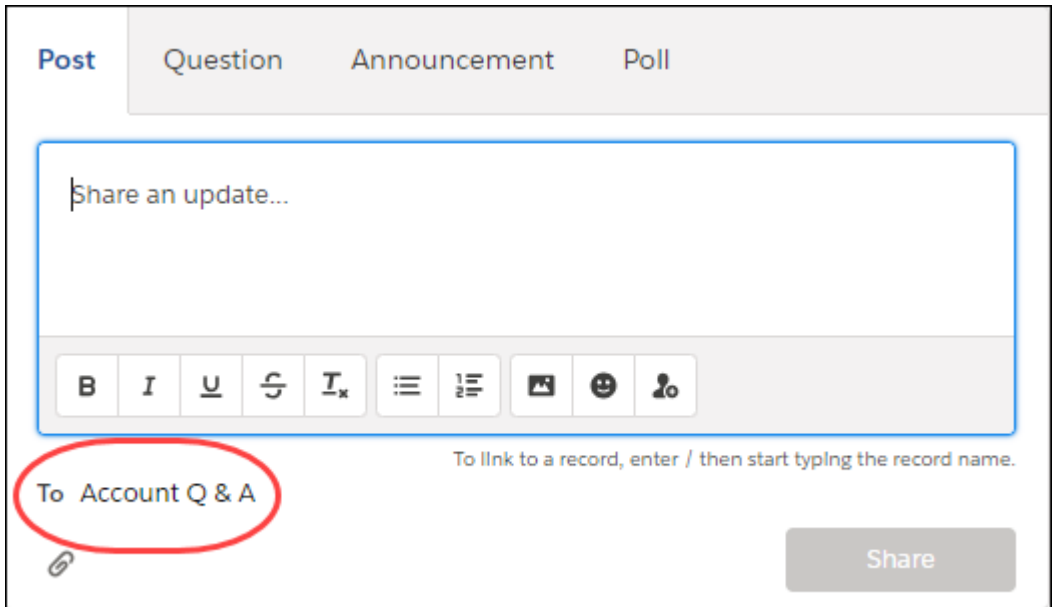


Your draft posts are visible to you and people with the View All Data permission (3).

Add files, links, images, mentions, hashtags, and any other rich text to your draft because we save it all. Well, we save all your posts. Questions and polls are not saved as drafts. When you save a post with a hashtag, the corresponding topic isn't created until the post is published.

### Little Things Mean a Lot

We changed the cursor focus when you click in the Chatter publisher. Old way: Focus goes to the To field. New way: Focus snaps your cursor to the text box. It saves the step of positioning your cursor, so you can start posting right away. It's also why we moved the To field below the text box. This focus change makes Chatter more accessible, particularly to people who use their keyboards to get around in Chatter.



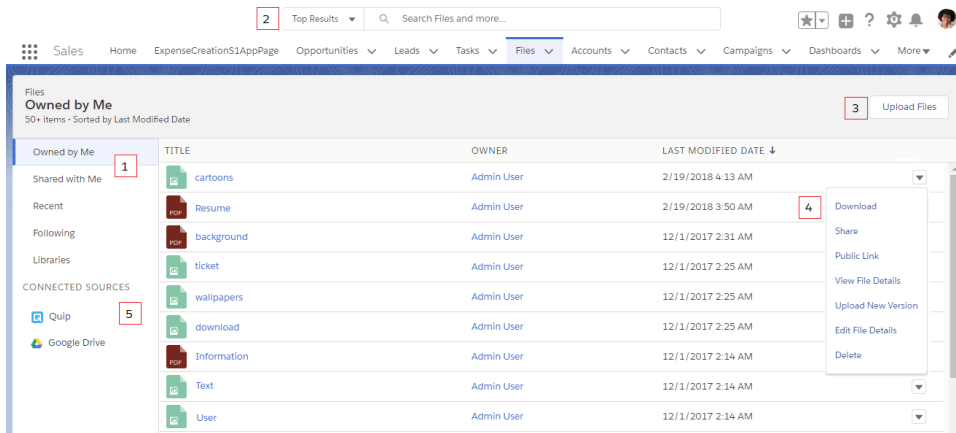
## File Management and Collaboration

Say goodbye to the days of juggling content across myriad types of Salesforce file management systems. In Lightning Experience, Salesforce Files unifies your users' files, documents, libraries, and attachments into a single place for easier file management. It also integrates access to files directly into the flow of your business, so users can get at the content they need from wherever they are in the app. Everyone in your org gets a welcome productivity boost with the integration of file sharing and content collaboration options into your sales and service processes.

## Home Base for Managing Files

Files home is a powerful environment for working with and managing files. All file types are supported, including everything from Microsoft® PowerPoint presentations and Excel spreadsheets to Adobe® PDFs and image files. Audio and video files can be uploaded and downloaded, but can't be previewed or played. (An exception is Lightning Dialer voicemail messages, which are saved as files and can be played. Lightning Dialer is available for an extra cost as an add-on license.)

## Chapter 9 Explore More of Lightning Experience



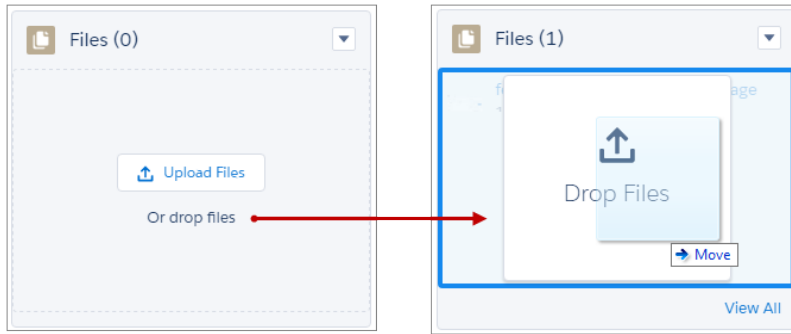
- Refine the list of files using filters (1).
- Access libraries and the files contained in libraries. Libraries are a way to organize content and share content with users (1).
- Search (2) to find files.
- Upload files to Salesforce Files (3).
- Act on files directly from the file list (4).
- If Files Connect is enabled for your org, the left navigation panel displays a section for external file sources, including Quip, Google Docs, and SharePoint (5).

Let's take a peek at a few of the other benefits your users get with Salesforce Files.

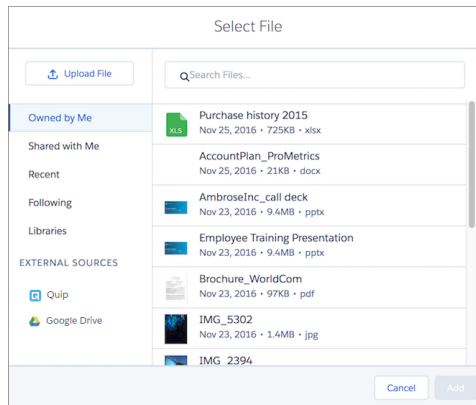
## Manage Files on Records and Groups

Have your users created attachments for records or groups using the old Notes & Attachments related list? With the new Files related list, you can give everyone a better way to manage related content.

When someone adds an attachment to a record, it exists only in the context of that record. Users can't get to the attachment from anywhere else in Salesforce. Pretty limiting, right? That's why we developed the Files related list. When you include the Files related list on an object's page layout, a user can upload files from their computer to attach to a record. The uploaded file is automatically added to Salesforce Files so it's accessible from anywhere, not just the record at hand. Simply drag a file—or multiple files at once—onto the related list! Or click **Add Files** > **Upload Files** to navigate through the file system to the right place.

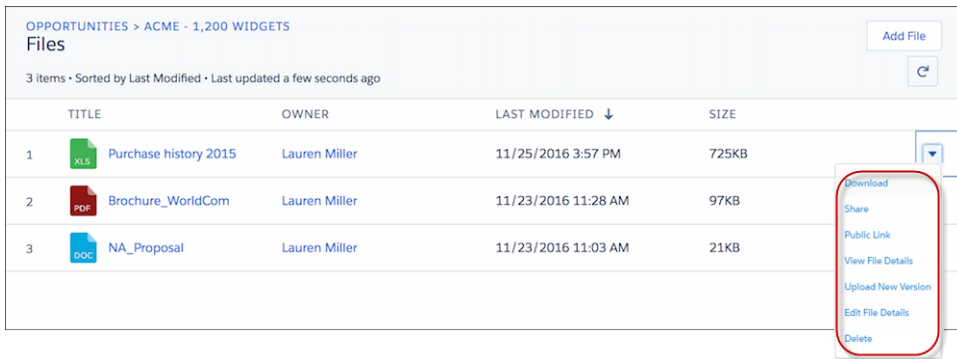


The Files related list also opens a door to all a user's files in Salesforce Files, including files in libraries and connected files in external repositories like Quip, Box, and Google Drive.






Drill in on the Files related list and all the rich features and flexibility of Salesforce Files are available for each file in the list. For example, upload a new version of a file, edit a file's details, or share a file with people, groups, and libraries.

## Chapter 9 Explore More of Lightning Experience



OPPORTUNITIES > ACME - 1,200 WIDGETS  
Files

3 items • Sorted by Last Modified • Last updated a few seconds ago

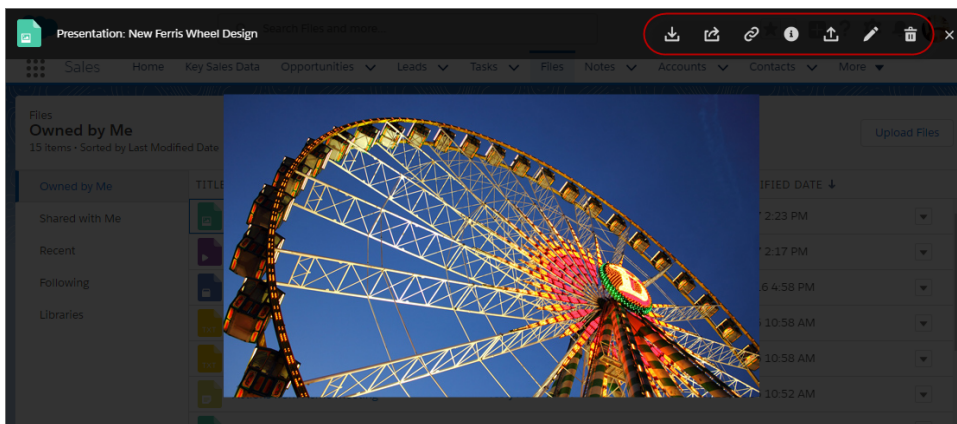
	TITLE	OWNER	LAST MODIFIED ↓	SIZE
1	 Purchase history 2015	Lauren Miller	11/25/2016 3:57 PM	725KB
2	 Brochure_WorldCom	Lauren Miller	11/23/2016 11:28 AM	97KB
3	 NA_Proposal	Lauren Miller	11/23/2016 11:03 AM	21KB

- Download
- Share
- Public Link
- View File Details
- Upload New Version
- Edit File Details
- Delete

## Preview Files

The beautiful file preview player in Lightning Experience provides a visually and functionally rich preview experience. Preview images are vector-based (rather than pixel-based), so the render quality is higher and doesn't degrade on high-resolution screens. And there's no need for your users to install Adobe Flash Player to preview files because the player isn't SWF-based.

The player includes controls for quickly downloading, sharing, or deleting the current file, uploading a new version of the file, editing file details, and generating a public link to the file. It also has full-screen mode and works with most presentation clickers, so reps can dazzle their audiences without needing to download anything.





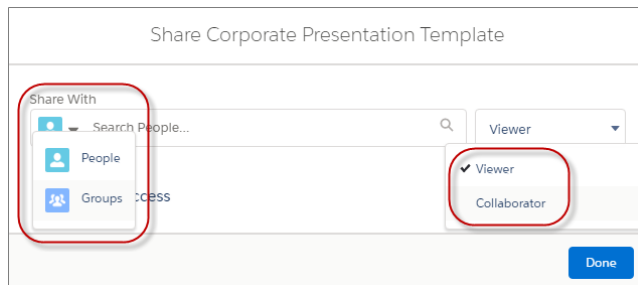
## Share Files

To facilitate content collaboration, Salesforce Files embeds file sharing opportunities throughout your company’s workflow. Users can access sharing options from Files home, Files related lists, the file preview player, file record detail pages, and Chatter feeds.

It’s easy to share files with the right audience.

### Specific people or groups

Share files with just the people or groups who should see them. Assign unique file permissions for each person or group. Viewers can view, download, and reshare the shared file (if the file allows resharing). Collaborators can do all that plus change file permissions, edit the file, and upload new versions.

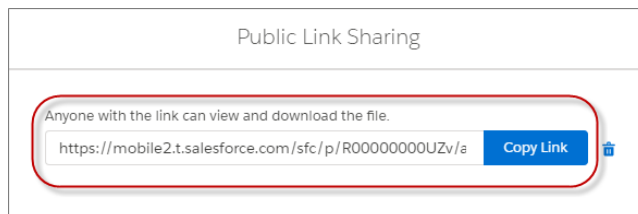


### Broader audiences or records

Attach files to posts or comments in the main Chatter feed or to feeds on records.

### Customers or people outside your company

Share a file with external audiences by creating a public link. A public link is an encrypted URL that you can send to people outside (or inside) your company, including leads, customers, partners, and coworkers. Recipients can view and download the file but they can’t be collaborators. You can delete the link at any time, at which point it no longer provides access to the file.



## Considerations for Other Content Types

Some older Salesforce content types aren't fully supported or supported at all in Lightning Experience. Here are a few things to keep in mind. See [Lightning Experience Considerations](#) in Salesforce Help for the full details.

### Attachments

Attachments on records aren't automatically available from Salesforce Files or the Files related list—because attachments aren't actually files. In Lightning Experience, users can continue to access attachments from the Notes & Attachments related list. But it's confusing to have content in two different lists on the same record. Fortunately you can convert attachments to files and eliminate the need for wasteful context switching. Install the [Attachments to Files tool](#) from AppExchange and kick off bulk jobs that convert attachments to Salesforce Files. Your users will thank you!

### Documents

The Documents tab isn't making the trek to Lightning Experience. But there are several options for making files from Documents available in Salesforce Files, including uploading them directly, using third-party data export tools, or using an API-based tool. See [Moving Documents to Salesforce Files](#) in Salesforce Help for more information.

One exception is documents used in email templates as logo headers and attachments. These files need to stay in Documents for now.

### Salesforce CRM Content

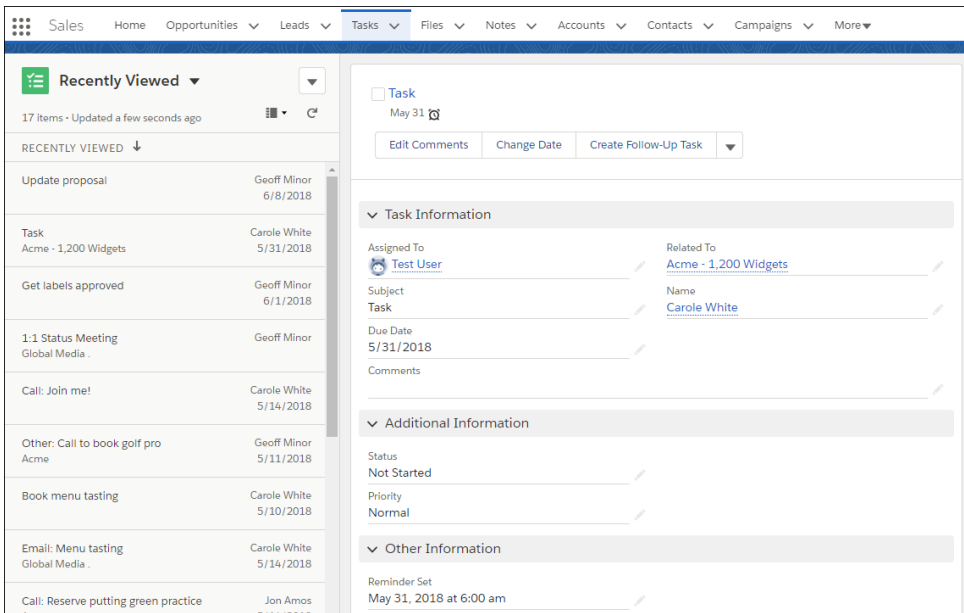
Folders and files in libraries are available from Salesforce Files in Lightning Experience. Users can share and interact with them as with other files. Library admins can create and manage libraries in Lightning Experience too. Content deliveries aren't available in Lightning Experience, but link sharing is supported.

## 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

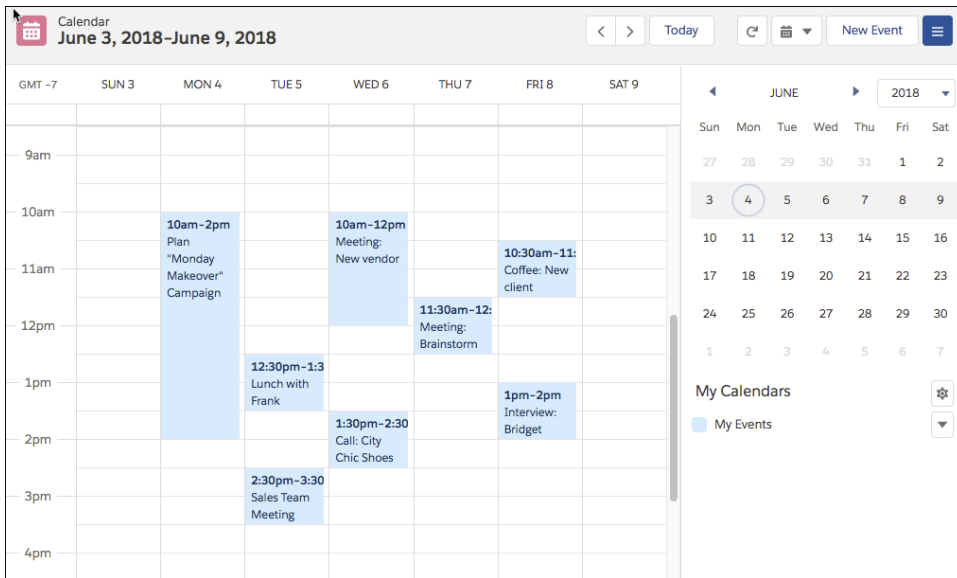


- View a list of all your open tasks, showing the opportunities, accounts, and other records they're related to. See and edit details right there on the same page.
- Switch to views of tasks you've delegated, open tasks, all overdue tasks, recently completed tasks, recently viewed tasks, recurring tasks, tasks due today, and unscheduled tasks.

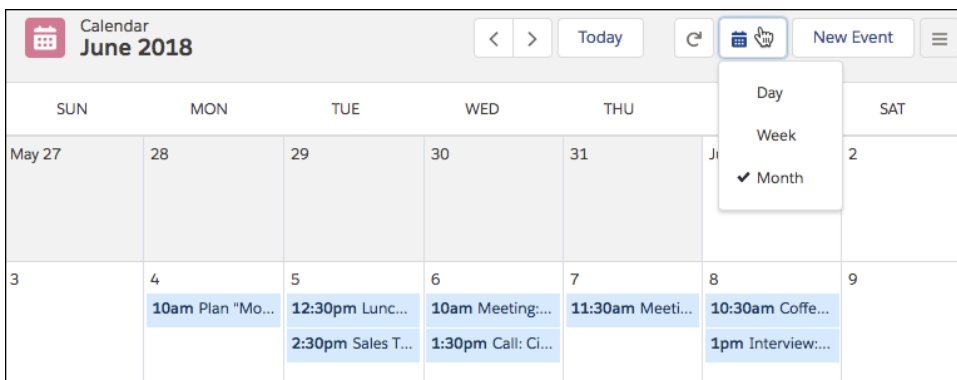
## Calendar

Here's the calendar week view:

## Chapter 9 Explore More of Lightning Experience

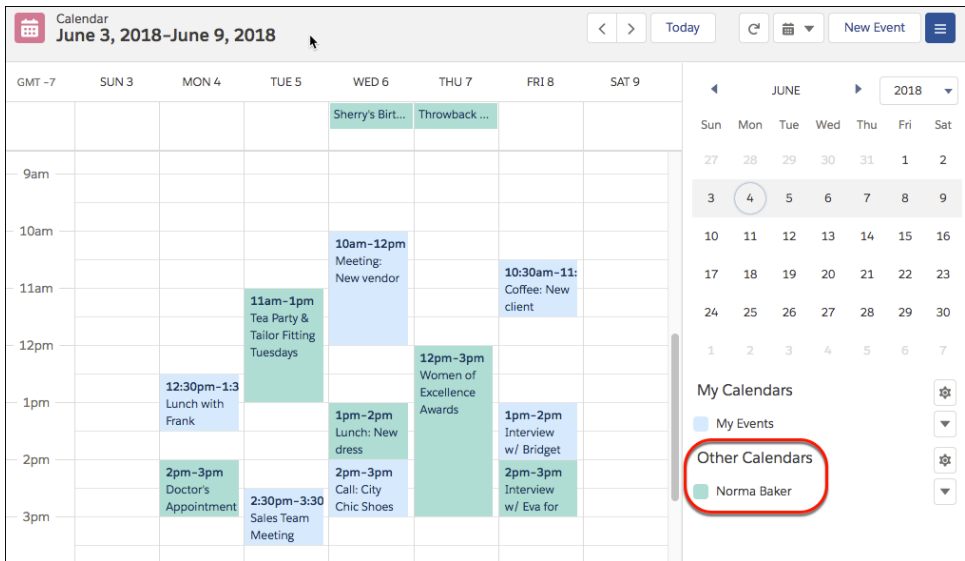


Or, change the calendar view to see your schedule for one day, or the whole month:

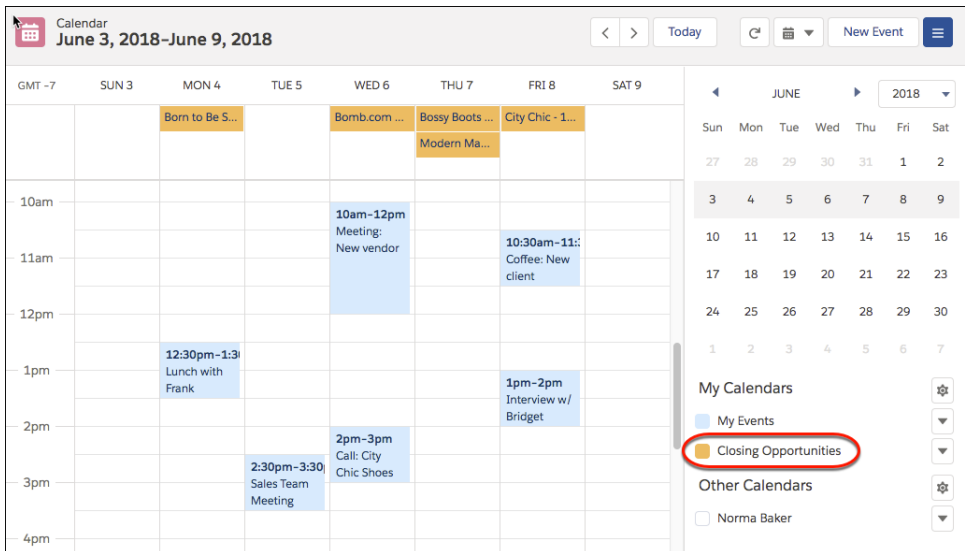


When you view, create, and edit events in Lightning Experience, the calendar is streamlined and easier to use. The calendar displays all events owned by a user, including events outside a user's business hours, in the time zone selected in your Salesforce settings.

To help you schedule meetings efficiently, you can take a peek at your coworkers' calendars, too. What you see depends on your admin-set sharing settings and the level of sharing access that coworkers give.



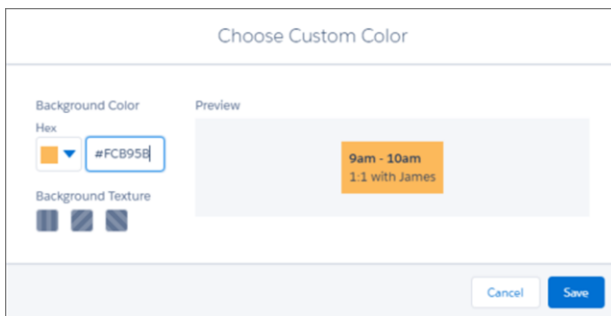
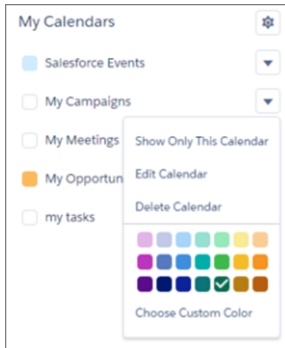
Let's say that you want to track campaigns, cases, retail store events, opportunities, and more. Just add calendars for all of these things. Choose a field to track on any standard or custom object. The calendar displays data in that field as calendar items.



## Chapter 9 Explore More of Lightning Experience

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.



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

To work with attendees and generate invitations from your meetings, ask your admin about [setting up the Lightning Sync feature](#).

# CHAPTER 10 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](#)

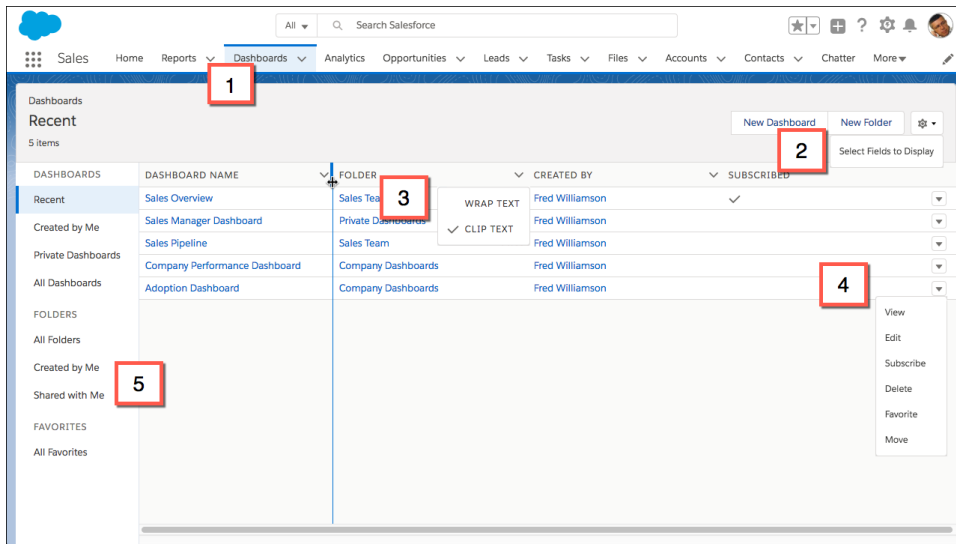
We've redesigned reports and dashboards for Lightning Experience, making them more interactive, easier to navigate, and easier to edit. Let's see what reports and dashboards in Lightning Experience can do!

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
- Without launching the report builder, you can edit report filters, show or hide a chart, totals, and details
- Find, manage, and create reports and dashboards more easily via redesigned home pages
- 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.



Access your reports from the Reports tab, and your dashboards from the Dashboards tab (1).

You can build a new report or dashboard (2) right from this page by clicking **New Report** or **New Dashboard**. Create a folder by clicking **New Folder**. Customize the tab layout by clicking the gear icon and then clicking **Select Fields to Display**.

Click a column heading (3) to sort by name, folder, creator, or whoever last modified a report or dashboard. Click again to reverse the sort order. Resize columns by hovering over the border area and dragging the boarder. Choose whether text in a column clips or wraps from the column actions menu.

The more actions menu provides click access to whatever you want to do (4). It's the fastest way to edit, subscribe, or favorite. But, if all you want to do is view a report or dashboard, it's a little quicker to click the name instead!

Find a report or dashboard using filtered lists and folders (5). 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.




# 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.

INDUSTRY	ACCOUNT ...	↑ OPPORTU...	AMOUNT	TYPE	LEAD SOUR...	EXPECTED R...
Apparel	Burlington T...	Burlington T...	\$235,000.00	New Custo...	Web	\$235,000.00
Electronics	Edge Comm...	Edge Comm...	\$75,000.00	New Custo...	Word of mo...	\$75,000.00

Get started from the toolbar (1). From here, you can add components and filters, undo or redo edits, edit chart properties, and perform administrative tasks like saving. Unique to Lightning Experience, from the chart properties menu, you can customize chart colors and dashboard theme (background color).

To edit an already-placed dashboard component, click  (2). You can choose a data-supplying source report, change chart types, edit the component title, and more. To remove a component, click **X**.

Arrange dashboard components more easily than ever by dragging them anywhere on the grid (3).

Lightning Experience dashboards now feature more than three columns. They feature up to nine columns and unlimited rows. Dashboard components can span as many columns and rows as you like. Drag the corners and sides of a dashboard component to make it bigger or smaller. Charts, tables, and metrics all dynamically resize. Go ahead and get creative with your dashboard layout!

## Chapter 10 Reports and Dashboards in Lightning Experience

Put a better table in your dashboards [4]. The Lightning table is a new dashboard component that's only available in Lightning Experience. How much better is it? Here's what it offers.

- Up to 10 columns and 200 rows!
- Columns come from source reports' report type. That means that you don't have to add a field to the source report to show it in the table.
- Show Chatter photos and conditional highlighting.
- Lightning tables are dashboard filter compatible.
- Customize dashboard color palette and theme.

Lightning tables don't appear when you view the dashboard in Salesforce Classic.

# Present and Share Information in Interactive Dashboards

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

The screenshot shows a Salesforce Lightning Experience dashboard titled "Sales Overview" with the following components and callouts:

- 1:** Dashboard header area containing the title "Sales Overview", the user name "Fred Williamson", and action buttons like "Follow", "Refresh", "Edit", and "Subscribe".
- 2:** A "View Report" link located below the "Opps Over Time" chart.
- 3:** A "View Report" link located below the "Opps by Stage" funnel chart.
- 4:** A tooltip for the "Value Proposition" segment of the "Opps by Stage" chart, showing a "Sum of Amount" of 58,700,000 (27.65% of 212M).
- 5:** A "View Report" link located below the "Opps by Owner" table.

OPPORTUNITY...	INDUSTRY	STAGE	TYPE	PROBABILITY (...)	AMOUNT
Hank Chen	Media	Prospecting	Existing Business	10%	\$70,000.00
Nadia Sm...	Manufacturing	Negotiation/R...	New Business	90%	\$500,000.00
Nadia Sm...	Manufacturing	Value Propositi...	Existing Business	50%	\$50,000.00
Brian Alis...	Manufacturing	Id. Decision M...	Existing Business	60%	\$40,000.00
Brian Alie	Technology	Closed Won	New Business	100%	\$250,000.00

View dashboards as other people **(1)**. If properties are set to allow you to choose whom you view the dashboard as, click **Change** to see the dashboard as someone else.

Refresh, edit, and collaborate **(2)**. Perform common tasks, like refreshing dashboard data, from the toolbar.

View source reports **(3)**. To view a component's source report, click **View Report**. When it opens, dashboard filters get applied to the source report.

Hover over charts to see more details **(4)**. Click a chart segment or datapoint to open a filtered source report.

Expand dashboard components to see a larger version of it **(5)**. If a chart is a little too small to make out an important detail, expand 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.

## Chapter 10 Reports and Dashboards in Lightning Experience

REPORT Pipeline By Industry								
Total Records	Average Employees	Total Amount			Total			
12	3,070	\$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.

## Chapter 10 Reports and Dashboards in Lightning Experience

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).

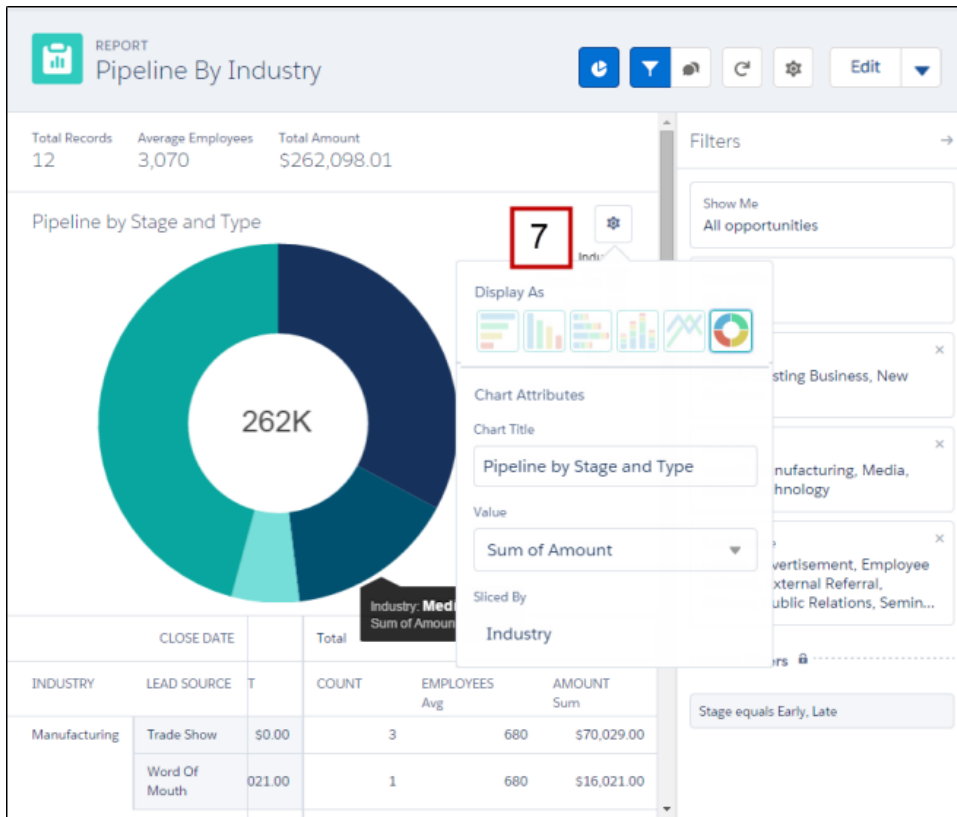
The screenshot displays a Salesforce report titled "Pipeline By Industry". At the top, key metrics are shown: Total Records (12), Average Employees (3,070), and Total Amount (\$262,098.01). Below this is a donut chart titled "Pipeline by Stage and Type" showing a total of 262K. A table below the chart provides a breakdown of the data. A filter menu is open over the donut chart, showing industry options like Agriculture, Apparel, Banking, Biotechnology, and Chemicals. The "Industry" filter is highlighted in yellow. A "Locked Filters" section is visible on the right, showing "Stage equals Early, Late".

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.

## Chapter 10 Reports and Dashboards in Lightning Experience



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.
- You can view and open folders that you created in Salesforce Classic in Lightning Experience. Lightning Experience obeys sharing rules set on report and dashboard folders in Salesforce Classic.
- Joined reports are available as a beta 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. Lightning tables, a table component only available in Lightning Experience, don't appear when you view a dashboard in Salesforce Classic.
- 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 about Lightning Experience reports and dashboards differences, see [Reports and Dashboards: What's Different or Not Available in Lightning Experience](#) in help.





# HOW YOUR EXISTING CUSTOMIZATIONS AFFECT LIGHTNING EXPERIENCE

## CHAPTER 11 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 page layouts 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 contains key record fields, and 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

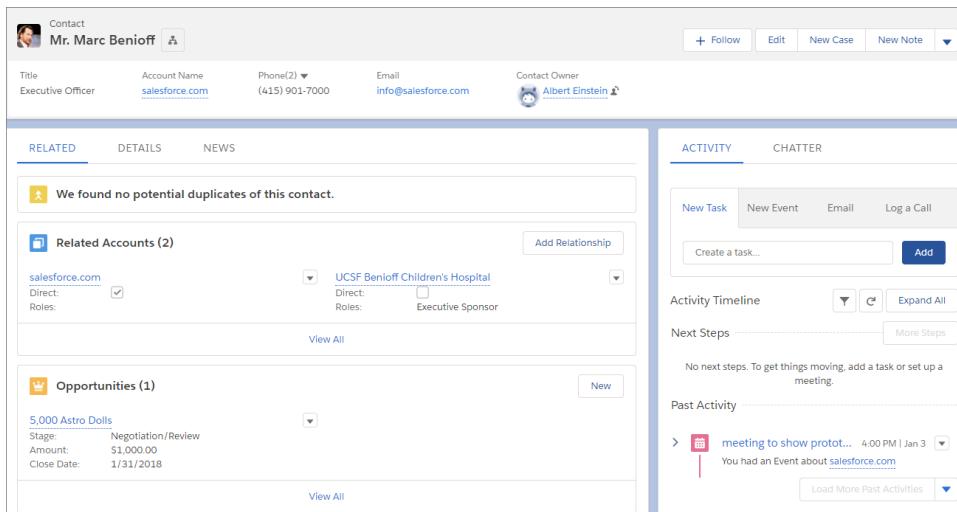
Available in: Lightning Experience

---

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

---

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



These page layout elements are supported in Lightning Experience.

## Actions

Actions display in different places, such as the highlights panel, Activity tab, and the Chatter tab. The actions are derived from the list of actions in the Salesforce Mobile 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 138.

### Blank Spaces

Blank spaces are supported in Lightning Experience.

### 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.

The top-down tab-key order, which allows users viewing a record detail page to move through a column of fields from top to bottom before moving focus to the top of the next column of fields, isn't supported in Lightning Experience or mobile. Even if a page layout is configured for a top-down tab-key order, tabbing moves from left-to-right through field columns in Lightning Experience and mobile.

### Related Lists

Related lists are included as Lightning components in Lightning Experience. Not all related lists are supported in Lightning Experience.

In Lightning Experience, only the first four fields on a related list are displayed. Users can click **View All** to see the rest of the fields.

### Report Charts

Report charts that you add to a page layout appear under the Details tab in Lightning Experience.

When you add a report chart to a page layout, it can take a few moments before the chart appears on Lightning record pages.

### Sections

Sections appear along with fields under the Details tab. A section with no header is incorporated into the section above it.

### Standard and Custom Buttons

Standard and custom buttons are treated as actions in Lightning Experience, just like in the Salesforce app.

 **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 Lightning Experience, Lightning Communities, and the mobile app** enabled display in Lightning Experience on Lightning pages, utility bars, and the Salesforce app.

## Chapter 11 Your Layouts Can Customize Lightning Experience Records

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.

- 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
- 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 Salesforce Mobile Basics module, you're familiar with compact layouts and how they work in the Salesforce mobile app. Compact layouts play the same role in Lightning Experience: displaying a record's key fields in the highlights panel of a record page. Compact layouts are also used in other places, such as the activity timeline and expanded lookup cards.

A compact layout lets you put the most important fields 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, up to the first seven fields in a compact layout appear in the highlights panel of an object record. (On smaller screens, the highlights panel displays fewer fields.) When a user hovers over a lookup relationship field on the object record page, a highlights panel for that field displays the first five fields from the compact layout. Highlights panels display the first field from the compact layout at the top in an accented font.

**OPPORTUNITY**  
**Burlington Textiles Weaving Plant Generator**

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

**Burlington Textiles C...**

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

**Opportunities (1)**

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

**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

ACTIVITY CHATTER

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

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

Event and task compact layouts determine the fields that show up in the details section when you expand an activity in the activity timeline in Lightning Experience. When you change the compact layout for tasks in the activity timeline, you also impact the fields that show up in the highlights area on tasks, in tasks lists, and everywhere else the compact layout is used.

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 Lightning Experience and the Salesforce app.



## CHAPTER 12 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 the Salesforce mobile app. In Lightning Experience, instead of showing up in one place—like the Chatter publisher or the Salesforce mobile app 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.

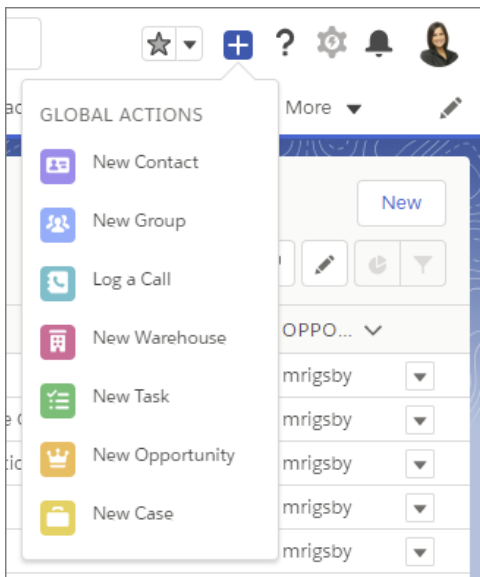
## Actions in Lightning Experience

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

 **Note:** Quick actions aren't the only action type covered here. Standard and custom buttons are also considered actions.

### Actions in the Global Actions Menu

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



The items in the menu appear in the order that they're listed in the Salesforce Mobile and Lightning Experience Actions section of the global publisher layout.

Actions associated with objects that aren't supported in Lightning Experience don't appear in the Global Actions menu. Also, the Global Actions menu doesn't support standard Chatter actions.

#### EDITIONS

Available in: both Salesforce Classic ([not available in all orgs](#)) 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 Canvas enabled), **Enterprise, Performance, Unlimited,** and **Developer** Editions

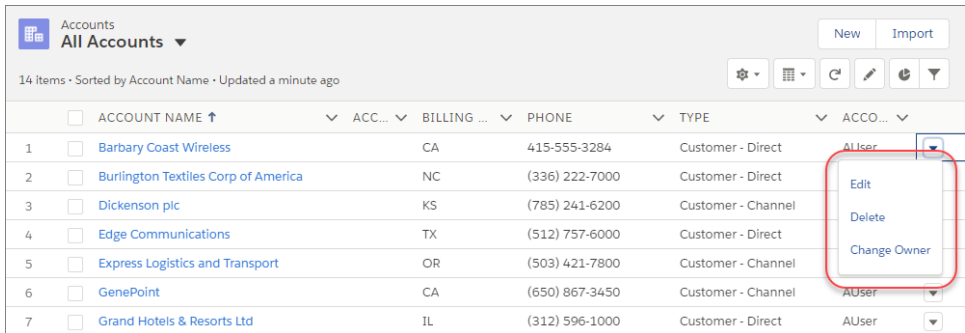


## Actions on List Views and List View Items

Custom buttons, list view actions, and certain standard buttons are supported on all list views, except Recently Viewed. To have a custom button appear on a list view, add the button to the object's List View search layout.

In the Kanban view, only the standard New action is supported.

List view items support only specific standard actions, like Edit, Delete, or Change Owner.



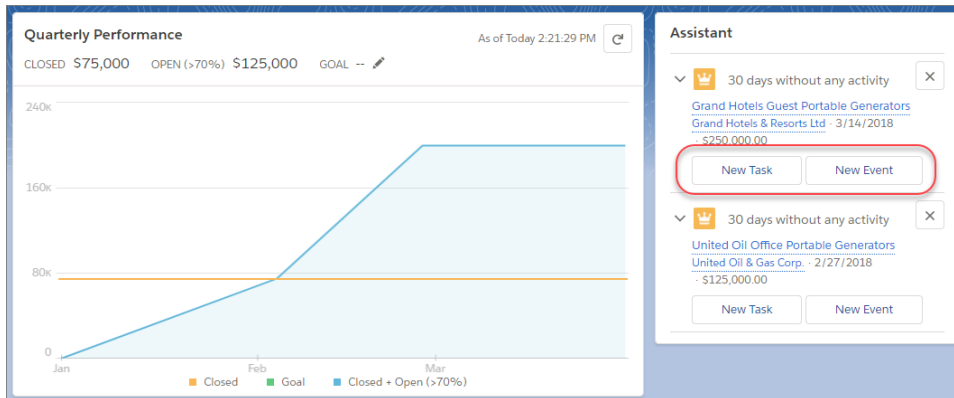
	ACCOUNT NAME ↑	ACC...	BILLING ...	PHONE	TYPE	ACCO...
1	<input type="checkbox"/> Barbary Coast Wireless		CA	415-555-3284	Customer - Direct	AUUser
2	<input type="checkbox"/> Burlington Textiles Corp of America		NC	(336) 222-7000	Customer - Direct	
3	<input type="checkbox"/> Dickenson plc		KS	(785) 241-6200	Customer - Channel	
4	<input type="checkbox"/> Edge Communications		TX	(512) 757-6000	Customer - Direct	
5	<input type="checkbox"/> Express Logistics and Transport		OR	(503) 421-7800	Customer - Channel	
6	<input type="checkbox"/> GenePoint		CA	(650) 867-3450	Customer - Channel	AUser
7	<input type="checkbox"/> Grand Hotels & Resorts Ltd		IL	(312) 596-1000	Customer - Direct	AUser

For Tasks, table format list views and the Kanban view support only standard buttons. Tasks list view items in table view and the task item detail pane in split view contain the complete list of available actions for tasks.

## Actions on the Home Page

On the Home page, you can find actions on recommendations in the Assistant. For example, imagine that 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.

## Chapter 12 Actions and Lightning Experience



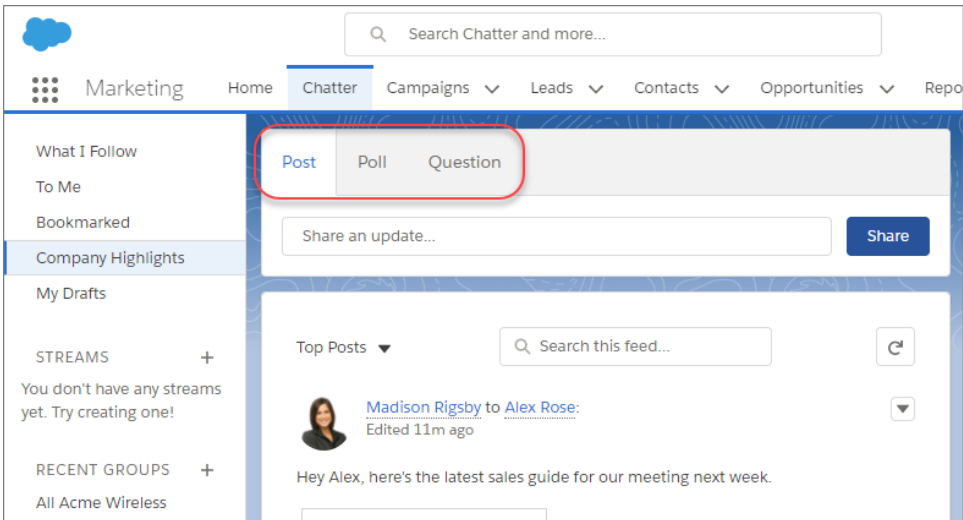
The actions that appear depend on the type of recommendation. To appear in the Assistant, actions must be added to the Salesforce Mobile and Lightning Experience Actions section of the global publisher layout. Supported actions include:

- New Task
- New Event
- Edit
- Email

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

## Actions on the Chatter Page

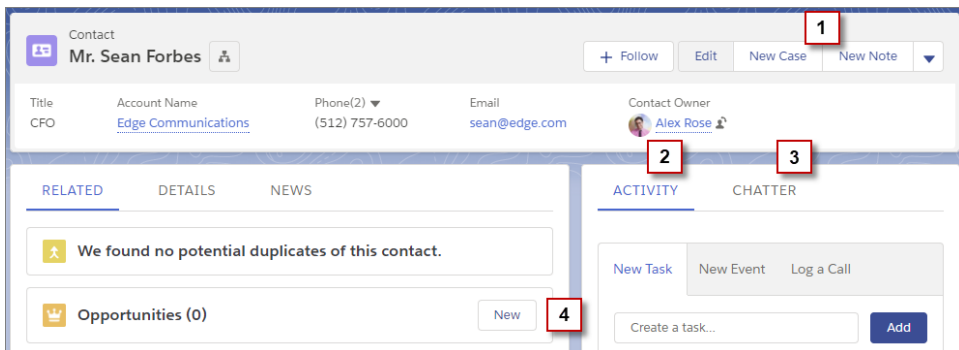
The Chatter page, like the Chatter tab on record pages, contains only standard Chatter actions. By default, only the Post, Poll, and Question actions are supported, and if you have Groups, the Announcement action. You can add, remove, or reorder the actions on the Chatter page from the Salesforce Mobile and Lightning Experience Actions section of the global publisher layout.



## 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.



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

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

## Chapter 12 Actions and Lightning Experience

- Standard buttons
- Custom object-specific Lightning component quick actions
- Custom flow actions (beta)
- Custom Visualforce quick actions
- Custom Visualforce buttons
- Canvas actions

The actions that appear in the page-level action menu are listed in the order that they appear in the Salesforce Mobile 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 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 user profiles, cases, and work orders can appear in a different way than on other records.


- Actions on the user profile page come from the Quick Actions in the Salesforce Classic Publisher section of the global publisher layout. Only standard Chatter actions appear on the user profile page, regardless of which actions are assigned to the User Page Layout or the global publisher layout.
- When feed tracking is enabled for cases or work orders, the page-level action menu on those records contains only custom buttons and supported standard buttons. Quick actions appear on the Chatter tab.

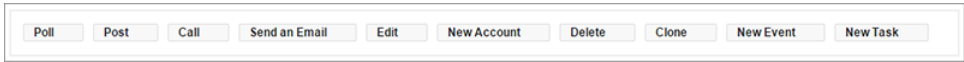
## Actions on Related Lists

Related lists in Lightning Experience (4) show custom list buttons and supported standard buttons assigned to the related list. Not all related list standard buttons are supported in Lightning Experience.

## Actions on Reports

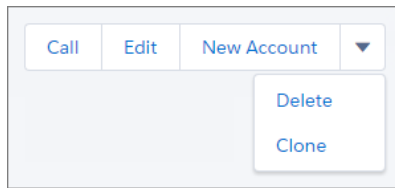
Actions on reports come from the Quick Actions in the Salesforce Classic Publisher section of the global publisher layout. However, only standard Chatter actions appear on reports, regardless of which other actions are assigned to the global publisher layout.

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

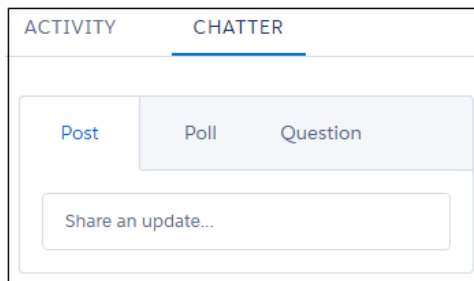


You have quick actions (New Account, New Event, New Task), a productivity action (Call), standard buttons (Edit, Delete, Clone, Send an Email), and Chatter actions (Poll, Post). Here's how those actions appear on a contact record page in Lightning Experience.

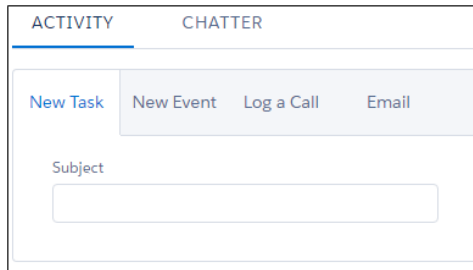
- The actions in the page-level action menu are a combination of the quick actions, productivity actions, and standard buttons. These actions appear in the order that they're listed on the page layout. Although they're quick actions, New Event and New Task don't show up here.



- 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 Salesforce Mobile 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 Salesforce Mobile 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

When you click to override the predefined actions in the Salesforce Mobile and Lightning Experience Actions section, the custom buttons in the buttons section of the page layout aren't automatically included in the action list. You must add the custom buttons as actions from the Mobile & 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.

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

### EDITIONS

Available in: both Salesforce Classic ([not available in all orgs](#)) 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 Canvas enabled), **Enterprise, Performance, Unlimited,** and **Developer** Editions

# CREATE LIGHTNING APPS

## CHAPTER 13 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.

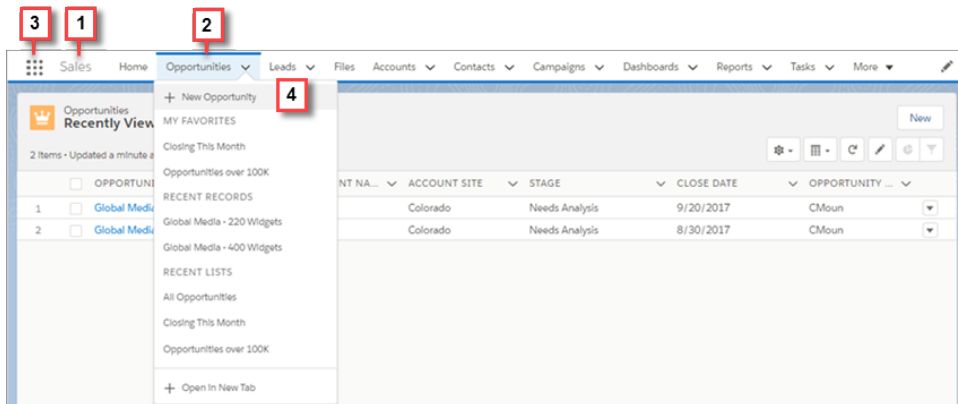
## Navigate at the Speed of Lightning

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
- Personalize the navigation bar to suit the unique way they work

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 home 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.

## How can users personalize the navigation bar?

Personalized navigation in Lightning Experience is similar to customized tab sets in Salesforce Classic but better. In Lightning Experience, the navigation bar can contain more than just object-level items, like Accounts. You can add granular items, like a dashboard, list, or record. Let's look at the different ways users edit the navigation bar.

You can drag items on the navigation bar to change their order.



**Note:** The asterisk to the left of the item name indicates that it's a temporary tab. A temporary tab opens when you open an item that doesn't have a parent object already in the navigation bar. The temporary tab is removed from the nav bar when you close it, log out of Salesforce, or switch to a different app.

Want to make several changes to the navigation bar? Click the pencil icon on the navigation bar. From the enhanced editor:

- Reorder the items already in your navigation bar.
- Rename items you've added. A pencil icon is next to the items that you can rename
- Add items to the navigation bar. Click **Add More Items**. Search through your favorites or all available items in your org, and choose what to add. After you make your selections, you can reorder or remove items before saving your changes. You can't delete items that your admin has specified for the app.

A few considerations to keep in mind.

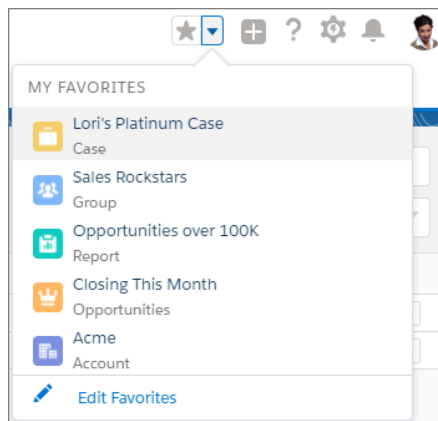
- Help users get the most out of personalized navigation by upgrading your Classic apps to Lightning apps. Users can't personalize the navigation bar of Classic apps in Lightning Experience.
- Check what's in your apps. Users can't remove the items you include in the navigation bar, and they can't personalize the navigation bar when it contains more than 50 items. For example, if you include 32 items in an app's navigation bar, users can add 18 more personal items.
- Items that you add to an app's navigation bar are added to the end of users' personalized navigation bars in the order that you added them.

## Chapter 13 What Is a Lightning App?

- When you remove an item from an app, that item remains in your users' personalized navigation bars, and users can then delete it.
- If you don't want your users to personalize the navigation bar for a specific app, disable personalization. From Setup in Lightning Experience, go to the **App Manager**. For the desired app, select **App Options**. Select **Disable end user personalization of nav items in this app**.
- If you don't want your users to personalize the navigation bar for any app, disable personalization. From Setup, enter *User Interface* in the Quick Find box, then select **User Interface**. Select **Disable Navigation Bar Personalization in Lightning Experience**. Salesforce recommends disabling navigation personalization by app instead of for the entire org.
- Control if temporary tabs are created when users access items outside of the app. From Setup in Lightning Experience, go to the **App Manager**. Edit the desired app. On the **App Options** page, select **Don't automatically create temporary tabs when users access items outside this app**.

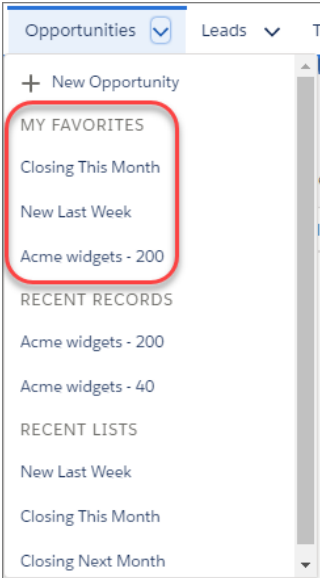
## Fast Track Navigation in Lightning Experience with Favorites

Favorites are another great way for your users to boost productivity by personalizing how they navigate. Think of the favorites star at the top of the page in Lightning Experience as a constellation in the sky. It's always there, so your users can easily navigate to what they need no matter where they are.



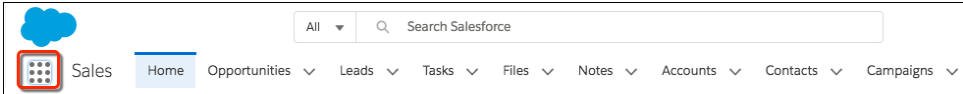
Click the favorites star to add the current page—for example, opportunities closing this month—to your favorites. Saved favorites are always just a click away in the favorites list. A highlighted star means you're on a favorite page. To remove a favorite, click the highlighted star. Your users can easily personalize their favorites lists by renaming favorites and rearranging items in the list.

If an item in the navigation bar has a dropdown menu, it automatically includes the user’s top favorites for the item.



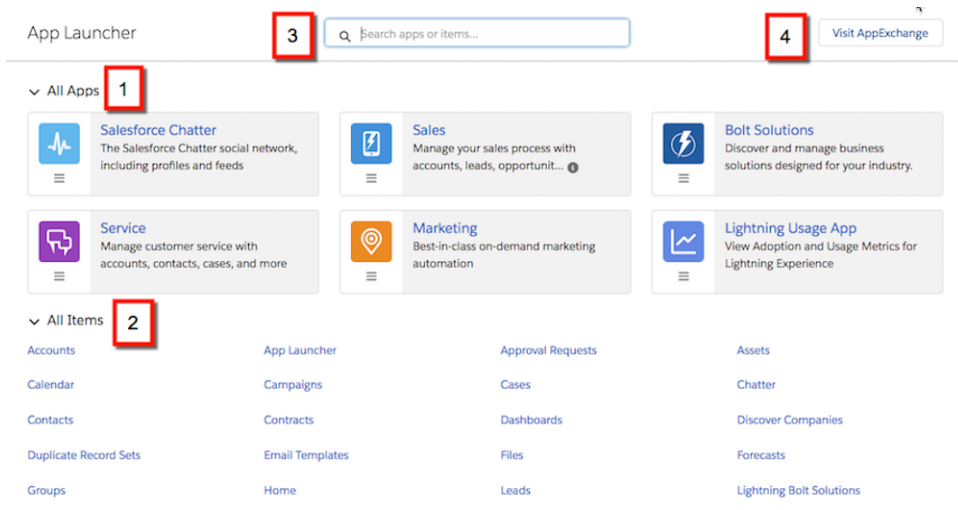
## Find Items and Apps in the App Launcher

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.

## Chapter 13 What Is a Lightning App?

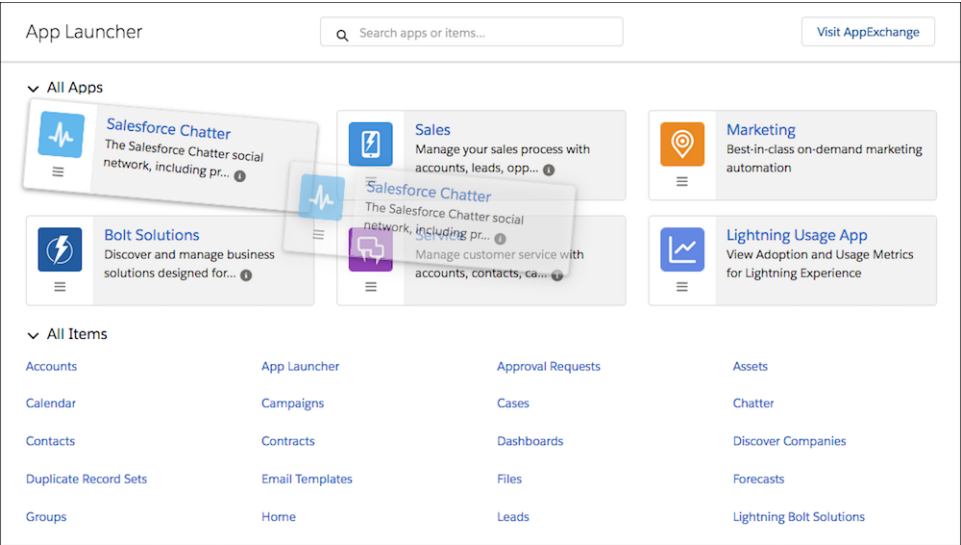


Apps show up in the App Launcher as large tiles under All Apps (1). Other items, such as custom objects, tasks, events, and the feed, show up under All Items (2). Don't see what you want? Search for it by name in the search box (3). Expand your search to find the latest cloud-computing apps and services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

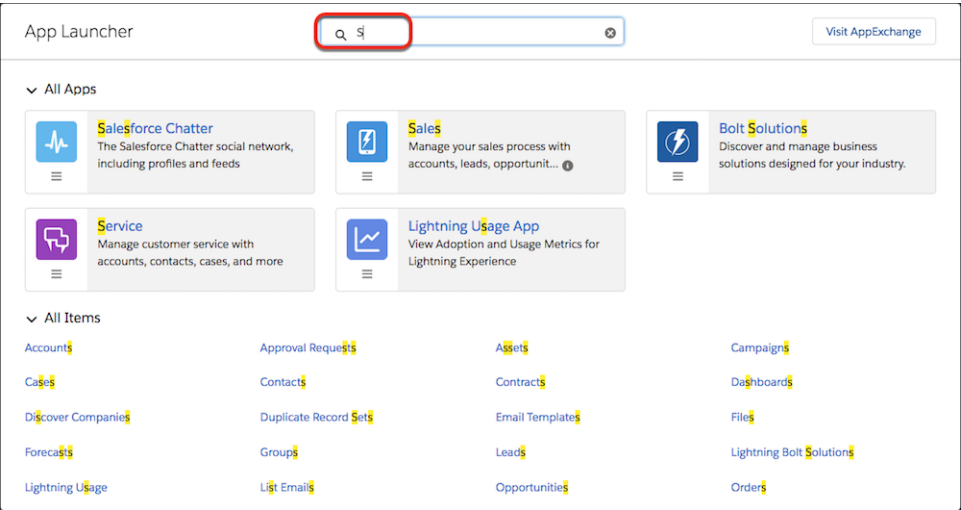
You can make various types of apps available to your users in the App Launcher, including:

- 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.



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.



## Meet the Lightning Experience App Manager

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

APP NAME	DEVELOPER NAME	DESCRIPTION	LAST MODIFIED DATE	APP TYPE	VISIBLE IN LIGHTNING EXPERIENCE
1 App Launcher	AppLauncher	App Launcher tabs	9/13/2017 1:54 PM	Classic	<input checked="" type="checkbox"/>
2 Community	Community	Salesforce CRM Communities	9/13/2017 1:54 PM	Classic	<input checked="" type="checkbox"/>
3 Content	Content	Salesforce CRM Content	9/13/2017 1:54 PM	Classic	<input checked="" type="checkbox"/>
4 Marketing	Marketing	Best-in-class on-demand marketing automation	9/13/2017 1:54 PM	Classic	<input checked="" type="checkbox"/>
5 Platform	Platform	The fundamental Force.com platform	9/13/2017 1:54 PM	Classic	<input checked="" type="checkbox"/>
6 Sales	Sales	The world's most popular sales force automation (SFA) solution	9/13/2017 1:54 PM	Classic	<input checked="" type="checkbox"/>
7 Sales	LightningSales	Manage your sales process with accounts, leads, opportunities, and more	9/13/2017 1:54 PM	Lightning	<input checked="" type="checkbox"/>
8 Sales Console	LightningSalesConsole	(Lightning Experience) Lets sales reps work with multiple records on one screen	9/13/2017 1:54 PM	Lightning	<input checked="" type="checkbox"/>
9 Salesforce Chatter	Chatter	The Salesforce Chatter social network, including profiles and feeds	9/13/2017 1:54 PM	Classic	<input checked="" type="checkbox"/>
10 Service	Service	Manage customer service with accounts, contacts, cases, and more	9/13/2017 1:54 PM	Classic	<input checked="" type="checkbox"/>
11 Service Console	LightningService	(Lightning Experience) Lets support agents work with multiple records across ...	9/13/2017 1:54 PM	Lightning	<input checked="" type="checkbox"/>
12 Site.com	Sites	Build pixel-perfect, data-rich websites using the drag-and-drop Site.com applic...	9/13/2017 1:54 PM	Classic	<input checked="" type="checkbox"/>
13 Warehouse	Warehouse		9/13/2017 2:39 PM	Classic	<input checked="" type="checkbox"/>

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 Experience” column mean?

A checkmark in the Visible in Lightning Experience 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 *Lightning Platform 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 *Salesforce Mobile App 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](http://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 to the Package Installation Details page.
5. Click **Install**.
6. Wait for the installation to finish, then click **Done**.
7. If your DE org is set to Lightning Experience, switch to Salesforce Classic.
8. From the app menu, select **Warehouse**.
9. Click the **Data** tab.
10. Click **Create Data**.
11. Click your name in the upper-right corner, then select **Switch to Lightning Experience**.





## CHAPTER 14 Step into the World of Lightning Apps

In this chapter ...

- [Create a Lightning App](#)
- [Create Apps for Different Audiences](#)

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


---

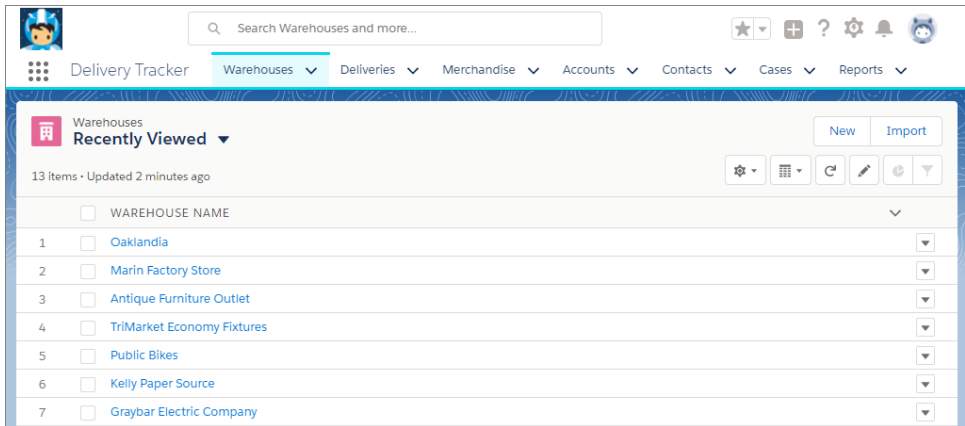
You've got the Warehouse app installed in your org. Let's create a mini app for the warehouse delivery managers, who want to know what inventory is getting delivered where, whether it got there on time, and what cases related to deliveries are open.

1. From the Home tab in 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! 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 .



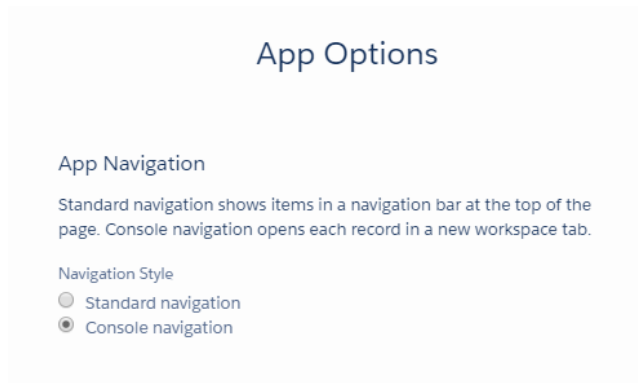
Nice work! Now you're ready to create your own custom Lightning apps.

### Beyond the Basics

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

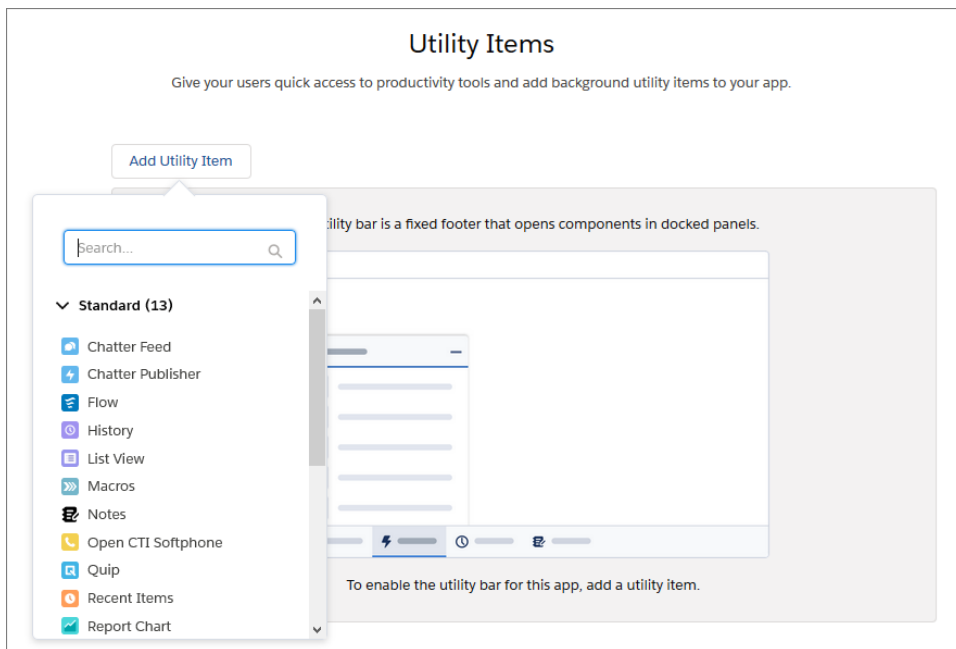
## Create Apps for Different Audiences

When you create a Lightning app, you get to decide which navigation style is best for your users and what important items belong in the app.



## Chapter 14 Step into the World of Lightning Apps

So which navigation style is best—standard or console? As with most things, it depends. Service users often work best using console navigation because it shows more information in a single view. Console navigation also provides quick access to standard and custom Lightning components, like integrated telephony, in an optional utility bar. And while console navigation is geared toward service users, other teams like marketing, development, and sales teams can also find it helpful. For example, sales teams can work from phone lead queues or get incoming leads through Omni-Channel, which are utility bar features available only in console navigation.



After deciding on the navigation style, your next step is to decide what belongs in the app.

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 15 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.




**Note:** You can't upgrade a Salesforce Classic console app to Lightning Experience. You can choose to display or hide the app in the Lightning Experience App Launcher, but you can't edit the app from the App Manager page in Lightning Experience Setup. To get started in Lightning Experience, customize these Salesforce-provided Lightning console apps: Service Console and Sales Console. You can also recreate your Salesforce Classic console app in Lightning Experience, but using Salesforce's out-of-the-box app is faster and easier.

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 the Home tab in 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.

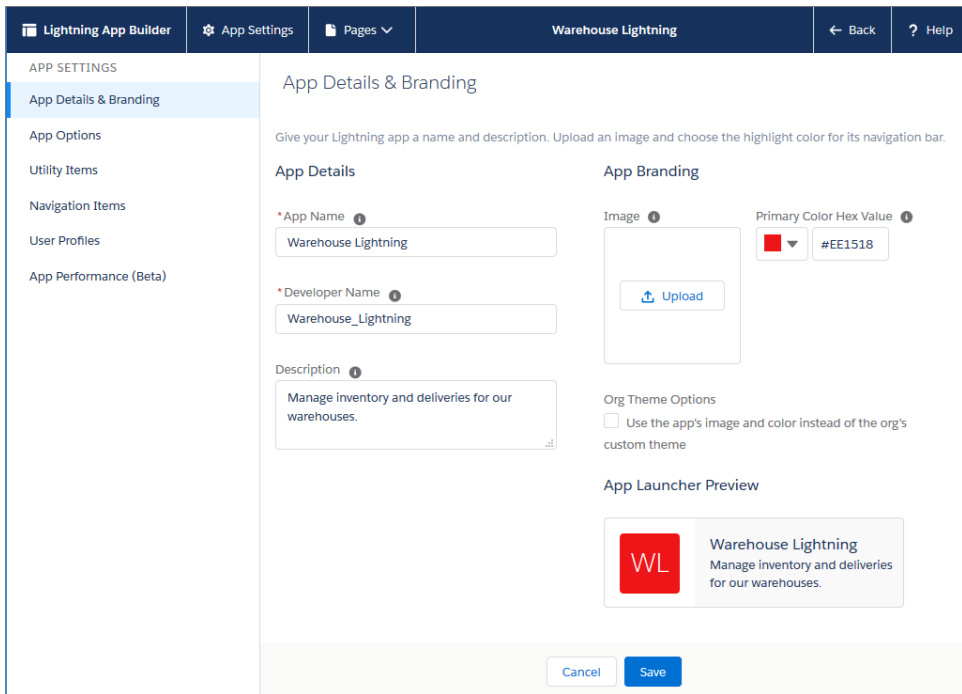
## Give the Warehouse App That Touch of Lightning

---

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.

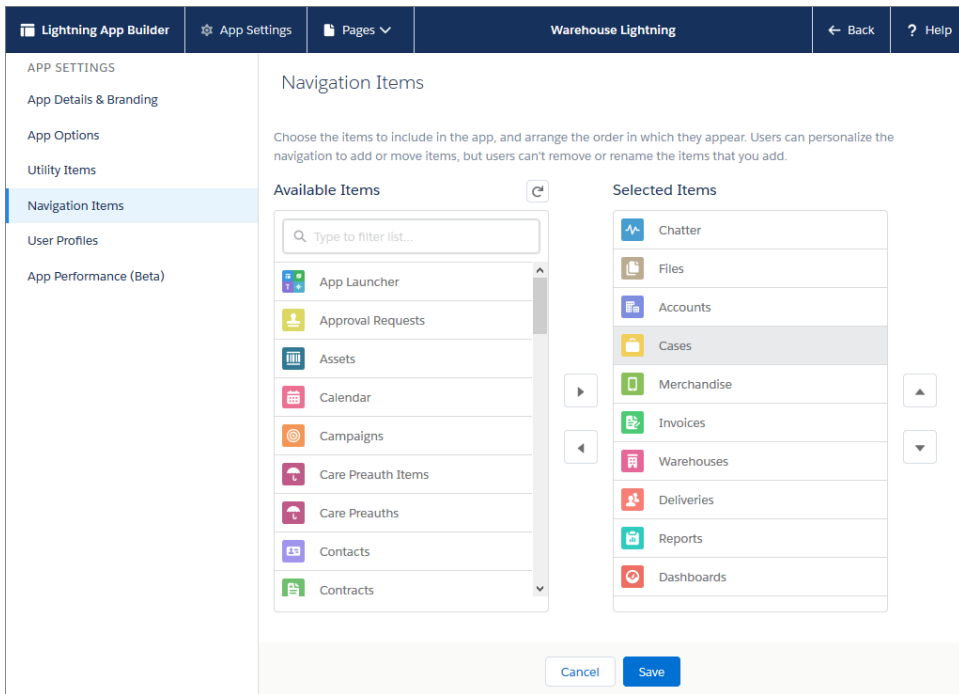
1. Find the Warehouse Lightning app in the list.
2. Click  from the Warehouse Lightning app row, and select **Edit**.  
The app opens inside the Lightning App Builder. Not only can you see all your app options, but you can view any active Lightning pages assigned to the app.
3. Update the description 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.






5. Click **Save**.
6. On the Navigation Items tab, 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.

## Chapter 15 Take Your Classic App to the Next Level: Lightning




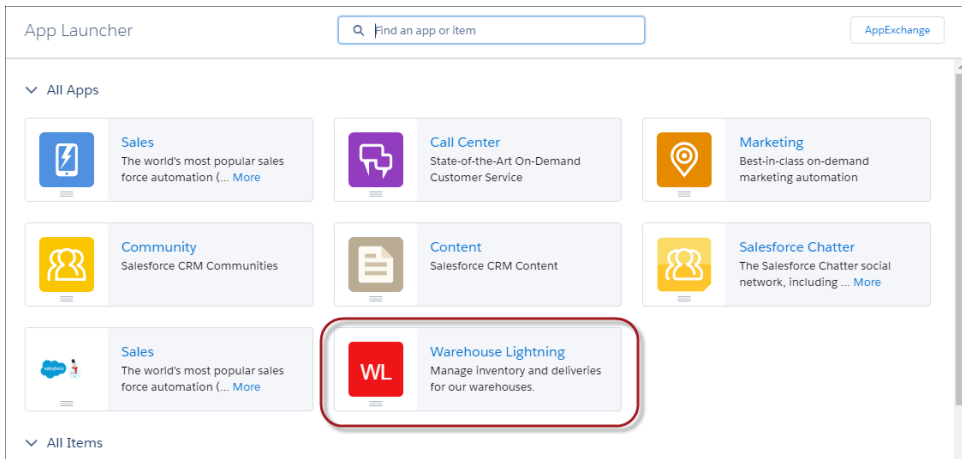
 **Tip:** If you have Lightning pages in your org, you can see them in the Available Items list. Being able to add Lightning pages to the navigation bar is another great perk about Lightning apps.

8. Click **Save**.
9. On the User Profiles tab, make sure that System Administrator is in the Selected Items list.
10. Click **Save**, and then click **Back** to return to Setup.

## 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.



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 16 Meet 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 the Salesforce mobile app and Lightning Experience, giving your users what they need all in one place.

But that's not all. When you edit a Lightning app from the App Manager in Setup, you're brought into the Lightning App Builder to manage the app's settings. You can update the app's branding, navigation, options, and manage the Lightning pages assigned to that app all in the Lightning App Builder.

In this module, we'll be exploring Lightning page creation. Check out the [Lightning Apps](#) module for more information about editing apps with the Lightning App Builder.

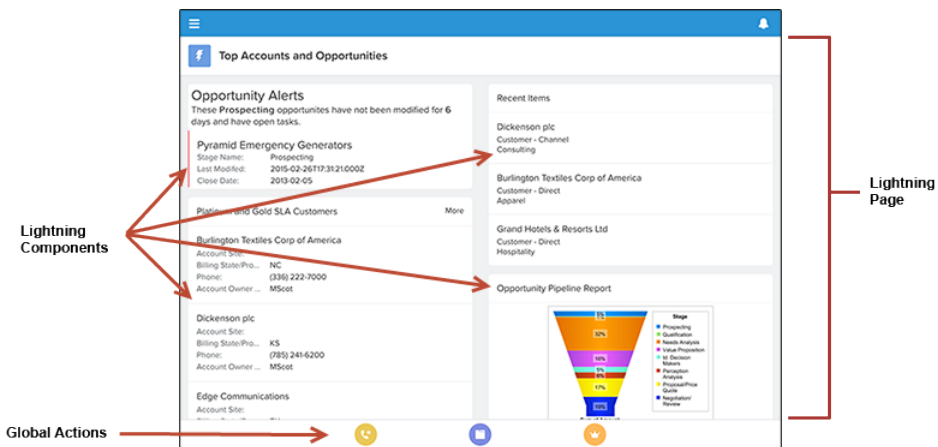
# How the Lightning App Builder Works

With the Lightning App Builder, you can build:

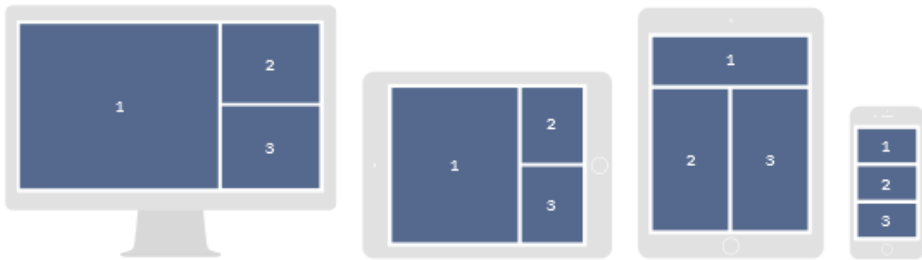
- Single-page apps that drill down into standard pages
- Dashboard-style apps, such as apps to track top sales prospects or key leads for the quarter
- “Point” apps to solve a particular task, such as an expense app for users to enter expenses and monitor expenses they’ve submitted
- Custom record pages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

A Lightning page is a custom layout that lets you design pages for use in the Salesforce mobile app or 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.



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.

Lightning pages support these components:

### 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

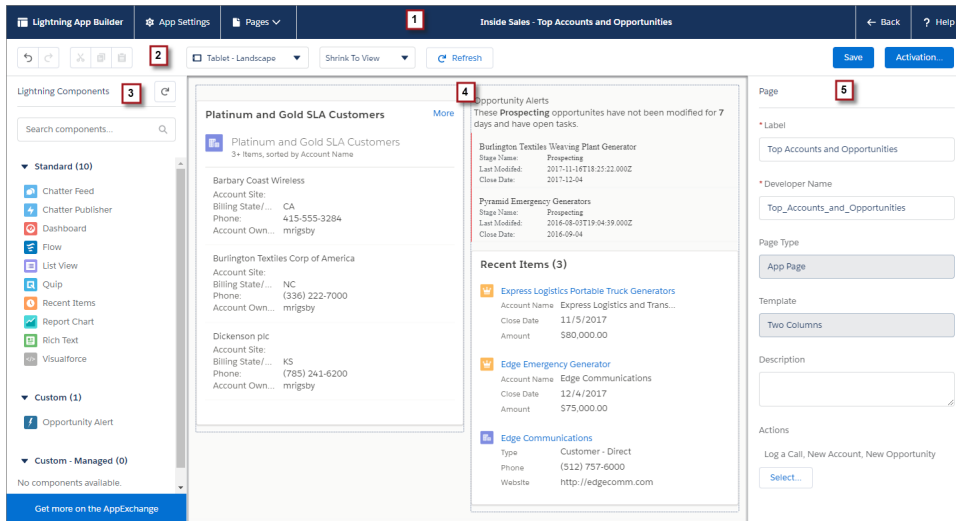
The AppExchange provides a marketplace for Lightning components. You can find packages containing components already configured and ready to use in the Lightning App Builder.

## 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.

## Chapter 16 Meet the Lightning App Builder



### Header (1)

When you're working on a Lightning page, the header shows you its label, and contains a Pages list where you can see the last 10 pages that you modified. You can also return to Setup without saving or to view more help for the Lightning App Builder.

If you're editing an app, the header also shows the app name and contains an App Settings tab where you can configure the app's options such as branding, navigation, and the utility bar. In an app context, the Pages list shows all active Lightning pages associated with the current app, including record and app pages assigned to the app, and all Home pages that are assigned to a profile.

### 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. In this module, we'll look at these three types.

### App Page

Use an app page to create a home page for a third-party app that you can add directly into the Salesforce mobile app 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.

Let's start with the Home page.

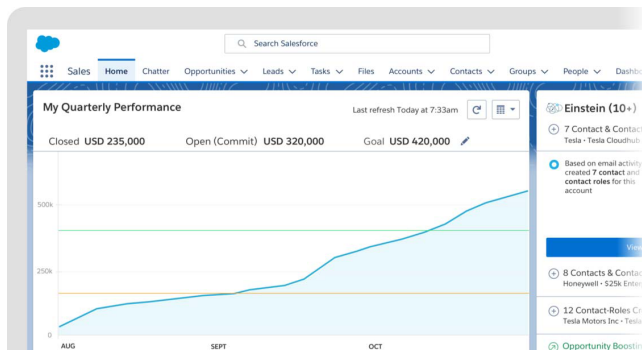


# CHAPTER 17 Yes, Virginia, You Can Customize Your Lightning Experience Home Page


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

## Chapter 17 Yes, Virginia, You Can Customize Your Lightning Experience Home Page

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

---

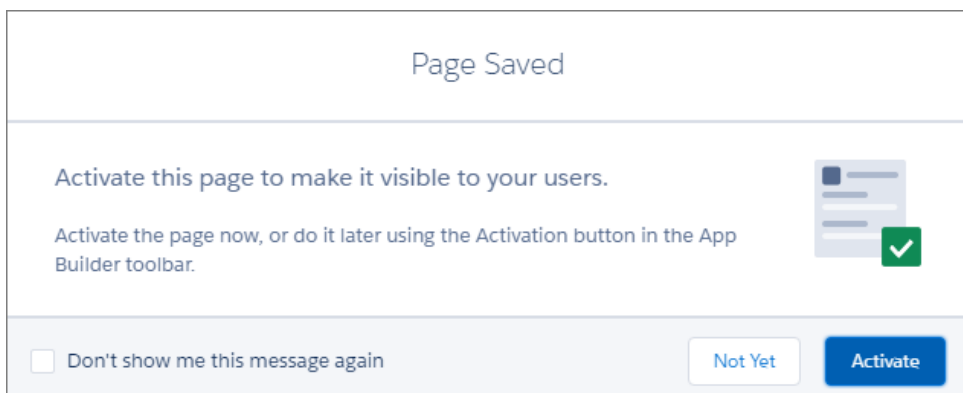
We'll tweak the position of the components on the standard Home page layout slightly to give you an idea of what's possible.

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 Performance to the top left region.
6. Add Today's Events to the lower left region and add Key Deals to the lower right region.
7. Add Today's Tasks above the Assistant.
8. Click **Save**.



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!

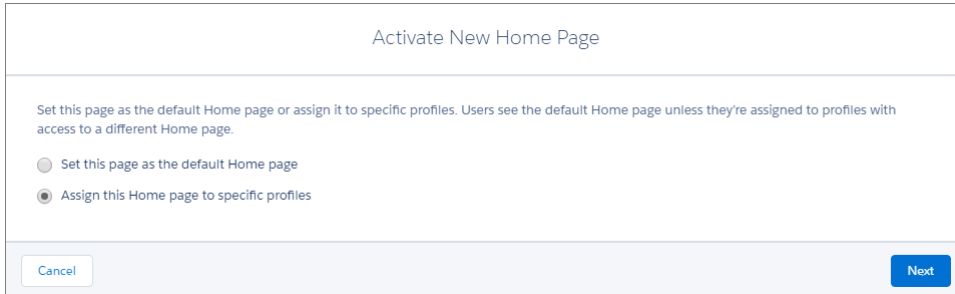
9. Click **Activate**, and we'll do just that.

 **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.



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**.

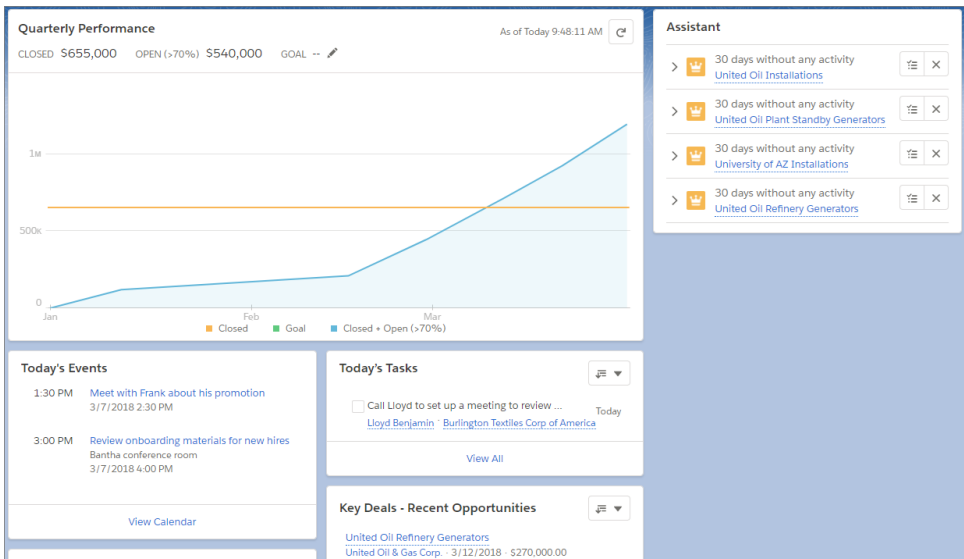
That's it. Let's go take a look.

## Look at Your Handiwork

---

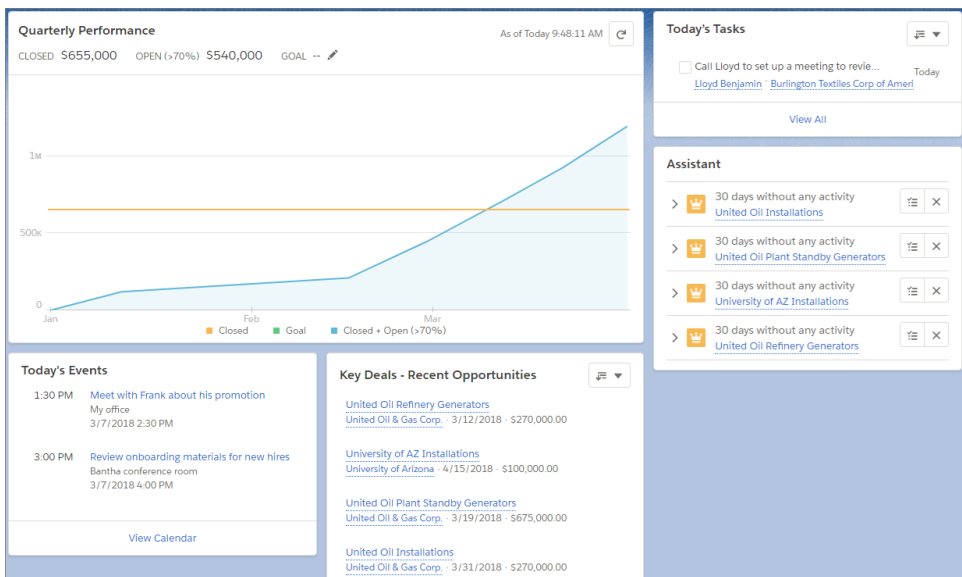
Just for reference, here's an example of the standard default Home page. Think of this as the "before" in this Home page makeover.

## Chapter 17 Yes, Virginia, You Can Customize Your Lightning Experience Home Page



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:



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

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

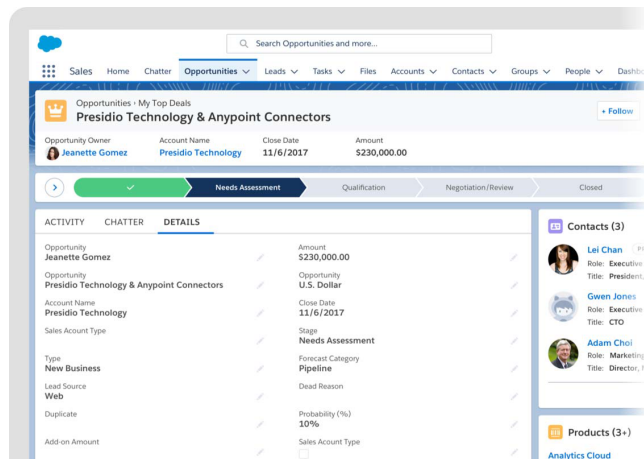


# CHAPTER 18 Those Lightning Experience Record Pages? You Can Customize Them, Too!

In this chapter ...

- Create a Custom Lightning Record Page
- Make Your Record Page Dynamic
- 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.



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

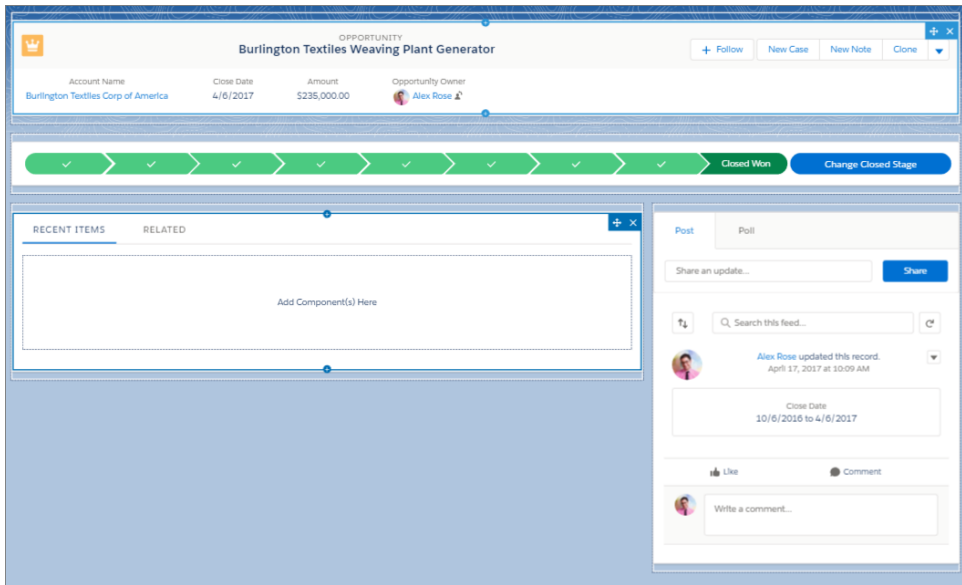
---

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.

## Chapter 18 Those Lightning Experience Record Pages? You Can Customize Them, Too!

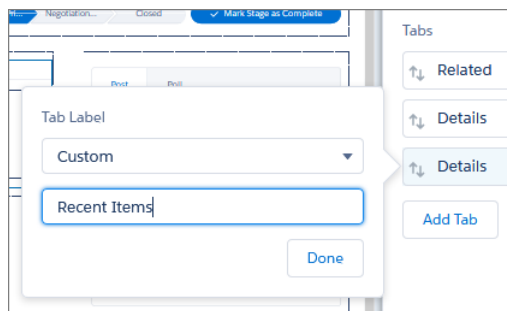


The Tabs component comes with some default tabs already in place. Let's add more.

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*.



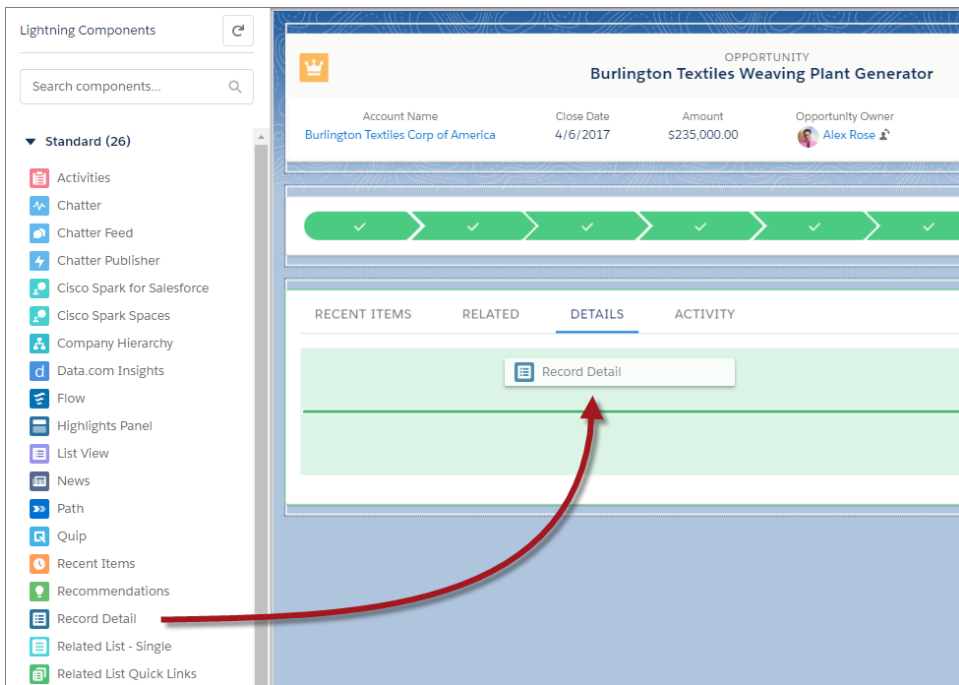
12. Click **Done**.

## Chapter 18 Those Lightning Experience Record Pages? You Can Customize Them, Too!

13. Create an Activity tab.
14. 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 a component to a tab, select the tab on the canvas and then drop a component directly below it.
15. Select the Details tab.
16. Drag a Record Detail component right below the Details tab, into the green highlighted area.



17. 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.
18. 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.

19. Click **Save**, then **Not Yet**.

## Make Your Record Page Dynamic

---

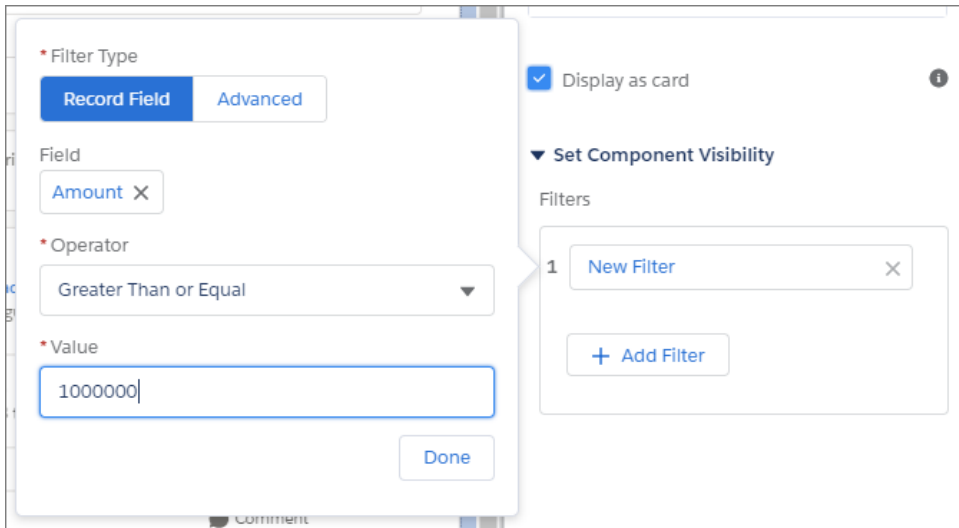
Did you know that you can control when a component appears on a Lightning record page? You can, by adding component visibility filter conditions and logic to its properties.

Component visibility properties appear when you select a component on a record, app, or Home page in the Lightning App Builder. This behavior applies to standard components, custom components, and components from AppExchange. No need to do anything to your custom components. It's all handled by the Lightning App Builder. If you don't define a filter, the component displays on the Lightning page as usual. When you define one or more filters and set the filter logic for a component, the component is hidden until the filter logic criteria are met.

Let's give it a shot. We'll construct filters to make a rich text component display when an opportunity's Amount is greater than or equal to \$1 million, and its Stage is Closed Won.

1. Add a Rich Text component above the Chatter component on the page.
2. Enter this text in the component: *A million dollar opportunity closed! Oh yeah!*
3. In the component properties, make the text bold and centered, 18-point size, and change the font to something that appeals to you.
4. Keep **Display as card** selected.  
This setting makes the text more readable on Lightning pages by adding a white background to it instead of a transparent one. Toggle the setting off and back on to see what we mean.
5. Click **Add filter**.
6. Set Field to **Amount**, if it's not already.
7. Set Operator to **Greater Than or Equal**. For Value, enter *1000000*.

## Chapter 18 Those Lightning Experience Record Pages? You Can Customize Them, Too!

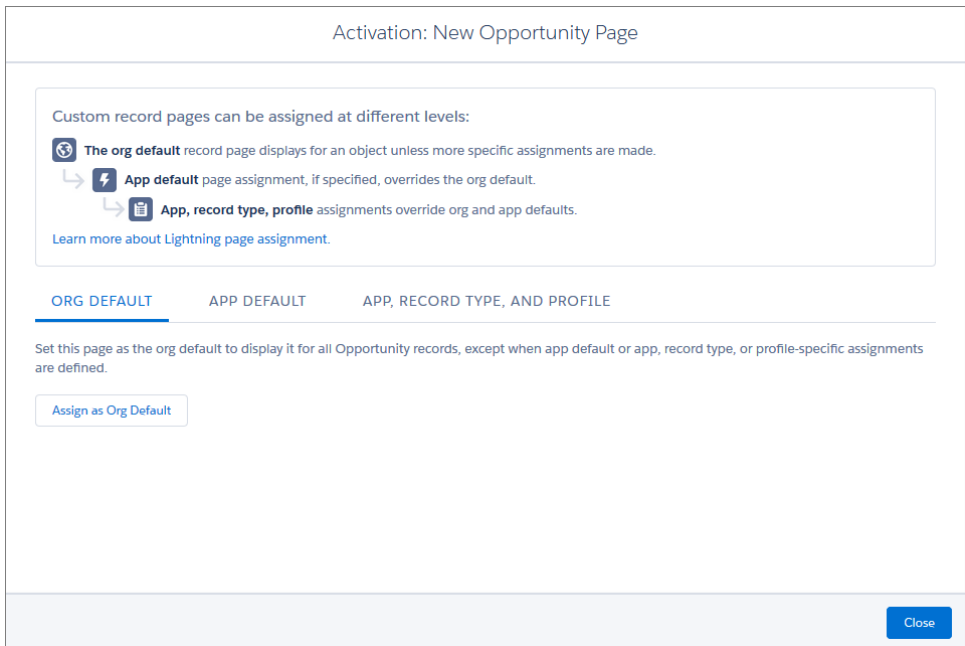


8. Click **Done**.
9. Click **Add filter** again, and create another filter for the Stage field equal to Closed Won.
10. Save the page again, then click **Activation**.

## 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.



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 deactivate my custom page?" That's easy, too. Click **Activation**, click the App, Record Type, 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 with an amount below \$1,000,000.

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 you 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.

The screenshot shows a Salesforce record page for an Opportunity. At the top, it displays the opportunity name 'United Oil Refinery Generators' and the account name 'United Oil & Gas Corp.'. Key fields include 'Close Date' (10/10/2017) and 'Amount' (\$270,000.00). The opportunity owner is Alex Rose. A progress bar shows stages: Proposal (selected), Negotiating, and Closed. Below the progress bar, there are tabs for 'DETAILS', 'ACTIVITY', 'RECENT ITEMS', and 'RELATED'. The 'DETAILS' tab is active, showing fields for Opportunity Owner (Alex Rose), Private (unchecked), Opportunity Name (United Oil Refinery Generators), Amount (\$270,000.00), Expected Revenue (\$202,500.00), and Close Date (10/10/2017). On the right side, there is a 'Post' section with a 'Share' button and a search bar for the feed. A recent feed item shows 'Alex Rose updated this record. Just Now'.

Don't see the rich text component that you added to the page? That's because the opportunity doesn't meet the criteria that you set. Let's change that.

4. From the page level actions in the Highlights panel, select **Edit**.
5. Change the opportunity's amount to be over \$1,000,000, change the stage to Closed Won, then click **Save**.

Woohoo, look at that! You didn't even have to refresh the page. As soon as you save your changes and the filter criteria are met, the page automatically updates to show the rich text component and its message.



## Chapter 18 Those Lightning Experience Record Pages? You Can Customize Them, Too!

The screenshot displays a Salesforce Lightning Experience record page for an Opportunity. The record is titled "United Oil Refinery Generators" and is owned by Alex Rose. The page shows a progress bar with several stages, the last of which is "Closed Won". A red arrow points from the "Closed Won" stage to a post in the right-hand feed. The post reads "A million dollar opportunity closed! Oh yeah!". Below the post is a search bar and a user profile for Alex Rose.

Account Name	Close Date	Amount	Opportunity Owner
United Oil & Gas Corp.	9/8/2017	\$1,270,000.00	Alex Rose

DETAILS	ACTIVITY	RECENT ITEMS	RELATED
Opportunity Owner Alex Rose		Amount \$1,270,000.00	
Private <input type="checkbox"/>		Expected Revenue \$1,270,000.00	
Opportunity Name United Oil Refinery Generators		Close Date 9/8/2017	
Account Name United Oil & Gas Corp.		Next Step	
Type Existing Customer - Upgrade		Stage Closed Won	

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

Let's move on to app pages.

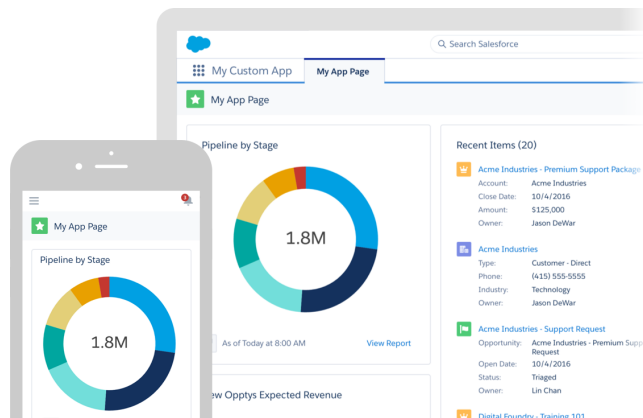


## CHAPTER 19 The App Home Lightning Page

In this chapter ...

- Create an App Page
- Add Your App Page to Lightning Experience and the Salesforce Mobile App
- Test Your App Page in the Salesforce Mobile App
- Test Your App Page in Lightning Experience

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



## Create an App Page

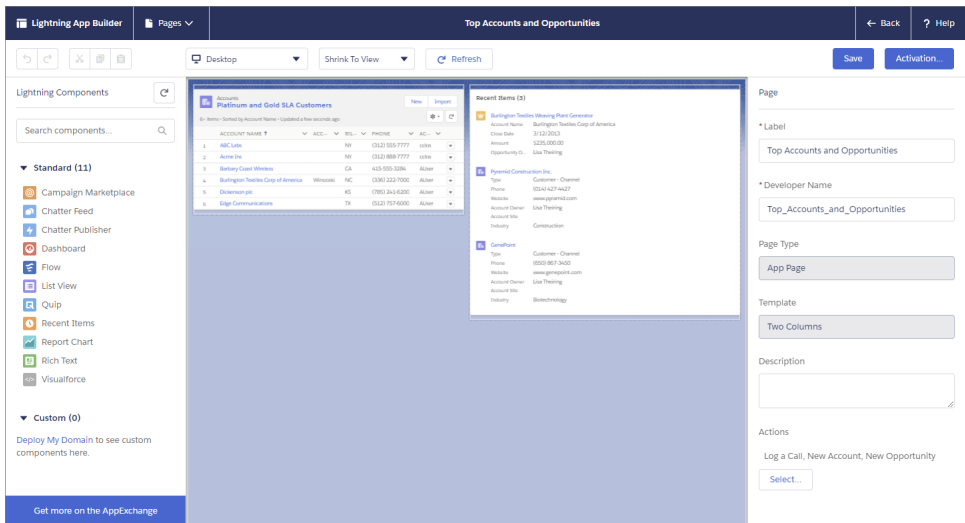
---

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.
6. Drag the List View component into the first region.
7. In the properties pane, under Filter, select the **Platinum and Gold SLA Customers** view.
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 OpportunityYou can add only global actions to a Lightning page.
11. From the toolbar, change the view to Tablet-Landscape.
12. Click **Save**, and then **Not Yet**.



Now that you've created your page, you're almost ready to add it to Lightning Experience and the mobile app.

## Add Your App Page to Lightning Experience and the Salesforce Mobile App

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 Salesforce mobile app navigation list and Lightning Experience app navigation bars, all in one place.

1. Click **Activation**.
2. Don't change the app name.
3. Change the icon to the blue lightning bolt icon.

The icon that you choose here is used as the icon for the mobile app and for the page in Lightning Experience.

4. Keep the tab's visibility open to all users.

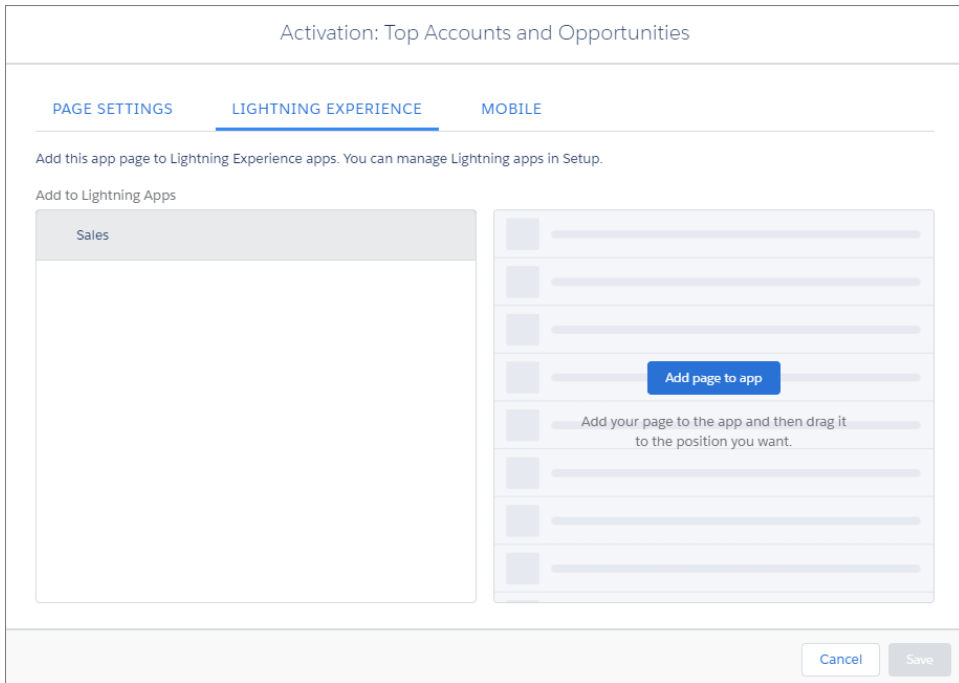


**Tip:** The **Activate for the System Administrator profile only** setting is useful while you're working on your Lightning page. Restricting your page to

## Chapter 19 The App Home Lightning Page

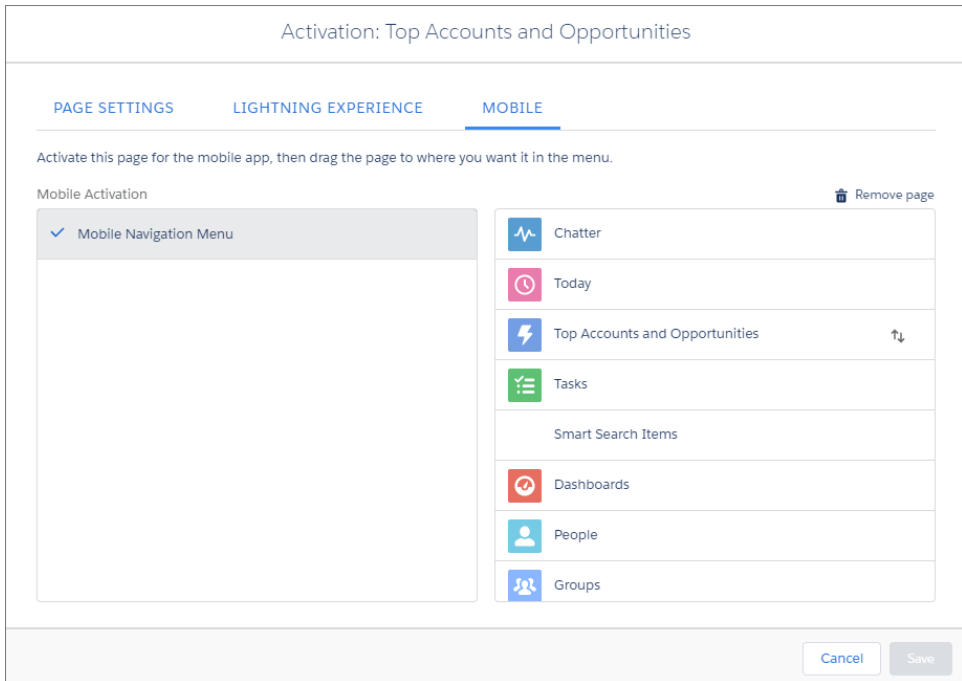
administrators only means that you can see and test the page, but your users can't see it until you're ready to expose it to them.

5. Click the **Lightning Experience** tab.
6. Select a Lightning app, and click **Add page to app**.



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 **Mobile** tab.
8. Select **Mobile Navigation Menu**, and click **Add page to app**.  
By default, new pages you add to the Salesforce mobile app 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 menu. We don't want that, so let's move it up.
9. Drag the page to below the Today menu item.




**10.** Click **Save**.

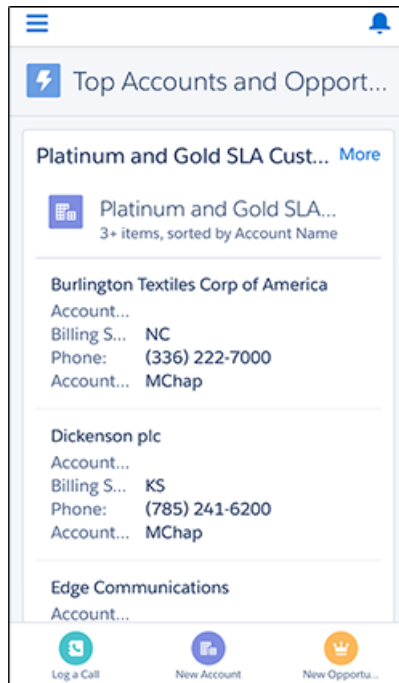
Your app home page is now ready for your mobile and Lightning Experience users!

## Test Your App Page in the Salesforce Mobile App

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

1. Open the Salesforce mobile app on your mobile device.
2. Log in using your Trailhead Playground credentials. (If you don't know your Trailhead Playground credentials, see [this article](#).)
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.



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 Trailhead Playground.
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.



The screenshot shows the Salesforce Lightning App Home page for the 'Top Accounts and Opportunities' app. The page features a top navigation bar with the Salesforce logo, a search bar, and user profile icons. Below the navigation bar, the app title 'Top Accounts and Opportunities' is displayed along with 'Log a Call' and 'New Account' buttons. The main content area is divided into two columns. The left column, titled 'Platinum and Gold SLA Customers', lists three customer accounts with their details. The right column, titled 'Recent Items (3)', lists three recent items with their details.

Customer Name	Account Site	Billing State/Province	Phone	Account Owner
Barbary Coast Wireless		CA	415-555-3284	Alias.mrigsby
Burlington Textiles Corp of America		NC	(336) 222-7000	Alias.mrigsby
Dickenson plc		KS	(785) 241-6200	Alias.mrigsby

Item Name	Type	Phone	Website	Account Owner	Account Site	Industry
Burlington Textiles Corp of America	Customer - Direct	(336) 222-7000	www.burlington.com	Madison Riggsby		Apparel
Dickenson plc	Customer - Channel	(785) 241-6200	dickenson-consulting.com	Madison Riggsby		Consulting
Burlington Textiles Weaving Plant Generator						



# CHAPTER 20 Custom Lightning Components in the Lightning App Builder

In this chapter ...

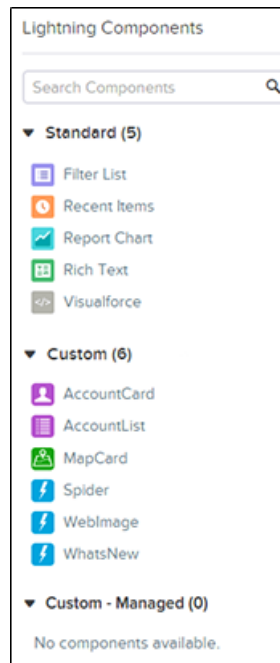
- [My Domain Is Already On in Your Trailhead Playground](#)
- [Install a Custom Lightning Component](#)
- [Add the Custom Lightning Component to Your App Page](#)
- [Test the Custom Component](#)

Creating a custom Lightning component requires programming skills. Or, you can install an existing Lightning component from the [AppExchange](#).

**!** **Important:** Point-and-click admins, you can do this unit! We're providing you with a custom component that's already configured. So don't worry. You don't need programming skills to earn this badge.

To learn how to create a custom Lightning component, see the [Lightning Components module](#).

Custom components in your org that are configured for use in the Lightning App Builder appear in the Lightning Components pane.



Your custom Lightning components don't automatically work on Lightning pages or in the Lightning App Builder. To make a custom component usable in both, you need to:

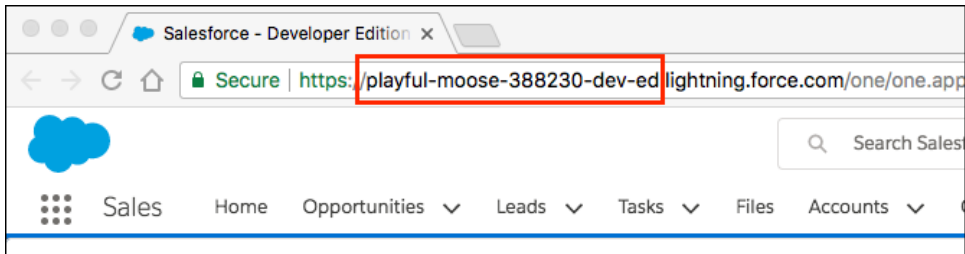
1. Configure the component and its component bundle so that they're compatible with the Lightning App Builder and Lightning pages.
2. Deploy My Domain in your org. When you deploy My Domain, references and links to Lightning resources are in the format `https://<myDomain>.lightning.force.com`.

The custom component we're providing for this module has already been configured for the Lightning App Builder.

## My Domain Is Already On in Your Trailhead Playground

---

Do not attempt to turn on My Domain, or change its settings, in your Trailhead Playground. By default, My Domain is already active in every Trailhead Playground.



In your production org, My Domain lets you create a subdomain unique to your organization. With My Domain, you replace the instance URL that Salesforce assigns you, such as `https://na17.lightning.force.com`, with your chosen subdomain, for example, `https://mydomainname.lightning.force.com`.

My Domain is required to create custom Lightning components and set up single sign-on (SSO) in an org. To learn more about My Domain, check out this [knowledge article](#). To learn how to activate it in your production org, see the [User Authentication](#) module.

## Install a Custom Lightning Component

---

We've provided a custom Opportunity Alert Lightning component that you can add to your Lightning page. Let's install it into your org.

1. Copy and paste `https://login.salesforce.com/packaging/installPackage.apexp?p0=04tj0000001mMYP` into the navigation bar of your browser, and then press ENTER.
2. Click **Install**.  
The installation takes a few moments.
3. Click **Done**.

After the installation completes, it's time to add the component to your page.

## Add the Custom Lightning Component to Your App Page

---

1. From Setup, enter *App Builder* into the Quick Find box, and then click **Lightning App Builder**.
2. Click **Edit** next to the Top Accounts and Opportunities page.
3. Add the OpportunityAlert component above the List View component.
4. In the properties pane, change `Days Since Last Modified` to `6`.  
Did you notice that the text inside the component changed to match the new field value?
5. Click **Save**.



**Tip:** Because you activated the page previously, you don't have to activate it again.

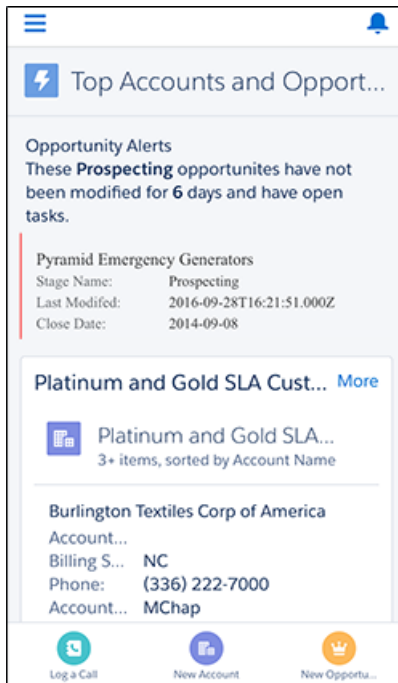
All right! The changes you made are immediately available in Lightning Experience and the mobile app. Let's see what the page looks like now.

## Test the Custom Component

---

We'll look at it in the Salesforce mobile app first.

1. Open the app on your mobile device.
2. If necessary, log in using your Trailhead Playground credentials.
3. Go to the Top Accounts and Opportunities page.



4. Go back to your org and view the Top Accounts and Opportunities page in Lightning Experience. Because you created the page using the Two Columns template, it uses the two-column format when you view it on the desktop or a tablet.

## Chapter 20 Custom Lightning Components in the Lightning App Builder

**Top Accounts and Opportunities** Log a Call New Account

**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** More  
3+ Items, sorted by Account Name

**Barbary Coast Wireless**  
Account Site:  
Billing State/Province: CA  
Phone: 415-555-3284  
Account Owner Alias: AUser

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

**Dickenson pic**  
Account Site:  
Billing State/Province: KS  
Phone: (785) 241-6200  
Account Owner Alias: AUser

**Recent Items**

**Pyramid Emergency Generators**  
Account Name: Pyramid Construction Inc.  
Close Date: 6/15/2016  
Amount: \$100,000.00  
Opportunity Owner: Alex Rose

**United Oil Refinery Generators**  
Account Name: United Oil & Gas Corp.  
Close Date: 8/5/2016  
Amount: \$270,000.00  
Opportunity Owner: Alex Rose

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

Congratulations! You used the Lightning App Builder to create an app page, personalize your Lightning Experience Home page, and customize a Lightning Experience record page.

You've got the skills now to make the Salesforce mobile app and Lightning Experience handier and more powerful for your users. Go forth and customize!



# MIGRATE JAVASCRIPT BUTTON FUNCTIONALITY TO LIGHTNING

## CHAPTER 21 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 built in the Cloud Flow Designer
- 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 the Salesforce mobile app and Lightning Experience.



## CHAPTER 22 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.

<b>JavaScript Button Top Use Cases</b>	<b>Lightning Alternatives</b>	<b><del>Deductive/Programmatic</del></b>
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

## Chapter 22 You Have Buttons? We Have Alternatives

<b>JavaScript Button Top Use Cases</b>	<b>Lightning Alternatives</b>	<b><del>Declarative/Programmatic</del></b>
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

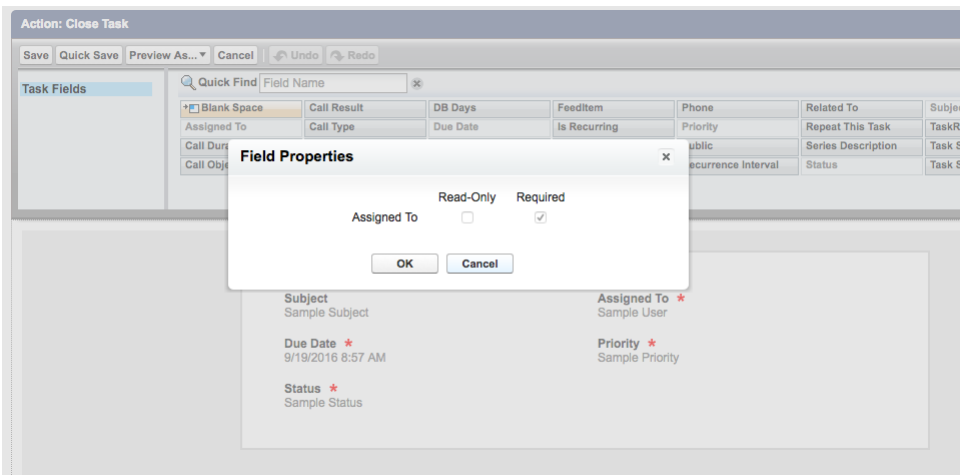
Sometimes you want to make sure that certain fields are filled in or populated with specific criteria when your users create or update records.

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. From the Object Manager in Setup, click **Task**, then click **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.



## Chapter 22 You Have Buttons? We Have Alternatives

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** Save Cancel

**Field Information**

Field	Task: Status
Field Type	Picklist

**Specify New Field Value**

A specific value |

Save Cancel

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 the Salesforce mobile app. Here's an example of a Close Task action on the Tasks page.

SALESFORCE

Search Salesforce

My App | Home | Chatter | Files | Projects | Leads | Accounts | Contacts | Cases | Opportunities | Reports | More

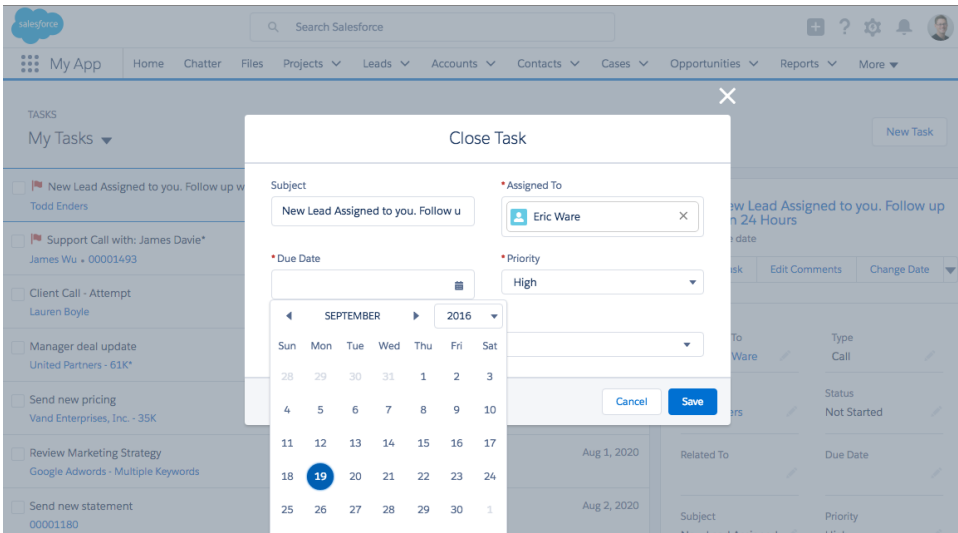
TASKS

My Tasks New Task

<input type="checkbox"/> New Lead Assigned to you. Follow up within 24 Hours Todd Enders	No due date	<input type="checkbox"/> New Lead Assigned to you. Follow up within 24 Hours No due date
<input type="checkbox"/> Support Call with: James Davie* James Wu - 00001493	Jul 28, 2020	<span>Close Task</span> <span>Edit Comments</span> <span>Change Date</span>
<input type="checkbox"/> Client Call - Attempt Lauren Boyle	Jul 28, 2020	Assigned To: Eric Ware   Type: Call
<input type="checkbox"/> Manager deal update United Partners - 61K*	Jul 30, 2020	Name: Todd Enders   Status: Not Started
<input type="checkbox"/> Send new pricing Vand Enterprises, Inc. - 35K	Jul 31, 2020	Related To:   Due Date:
<input type="checkbox"/> Review Marketing Strategy Google Adwords - Multiple Keywords	Aug 1, 2020	

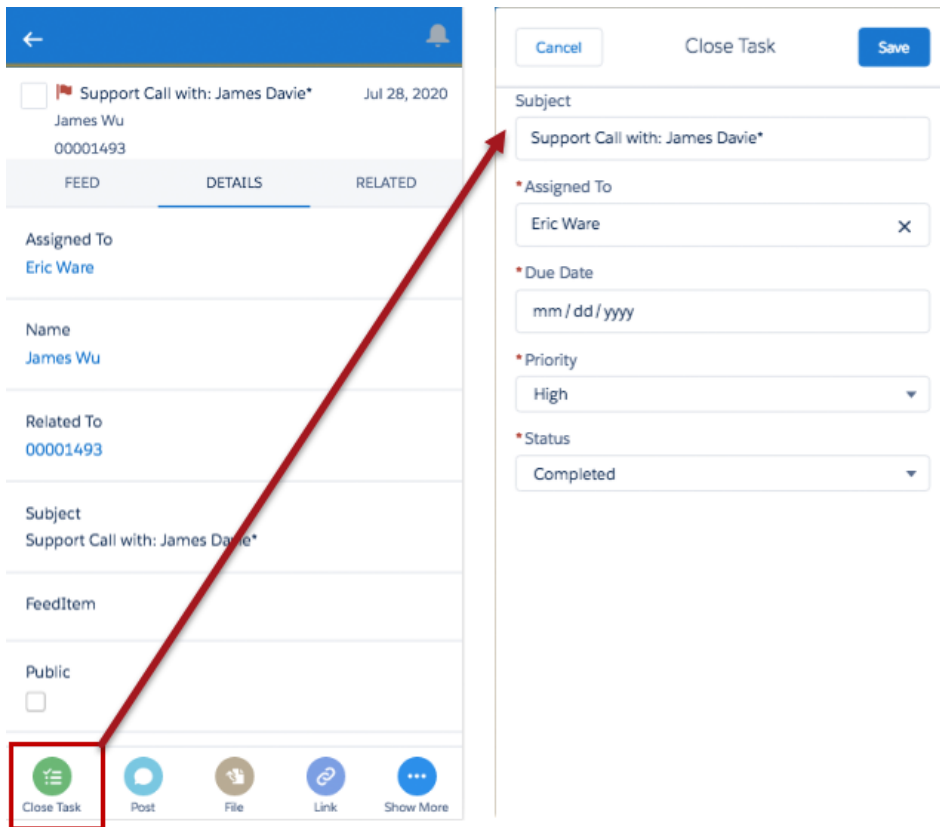
Clicking **Close Task** brings up the action, which the user can quickly act upon and save.





Here's what it looks like in the Salesforce mobile app.

## Chapter 22 You Have Buttons? We Have Alternatives



### 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.

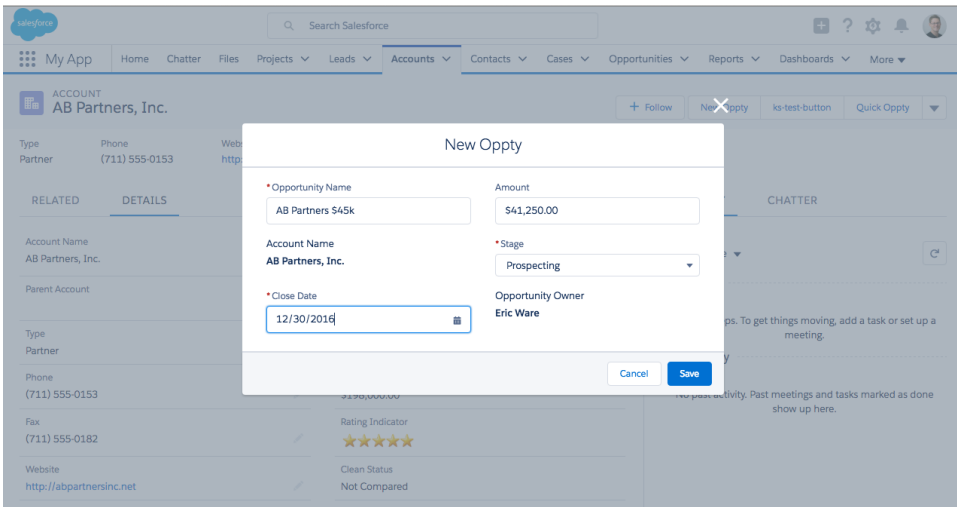
1. From the Object Manager in Setup, click **Account**, then **Buttons, Links, and Actions**.
2. Click **New Action**.
3. For Action Type, select **Create a Record**.

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.



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.

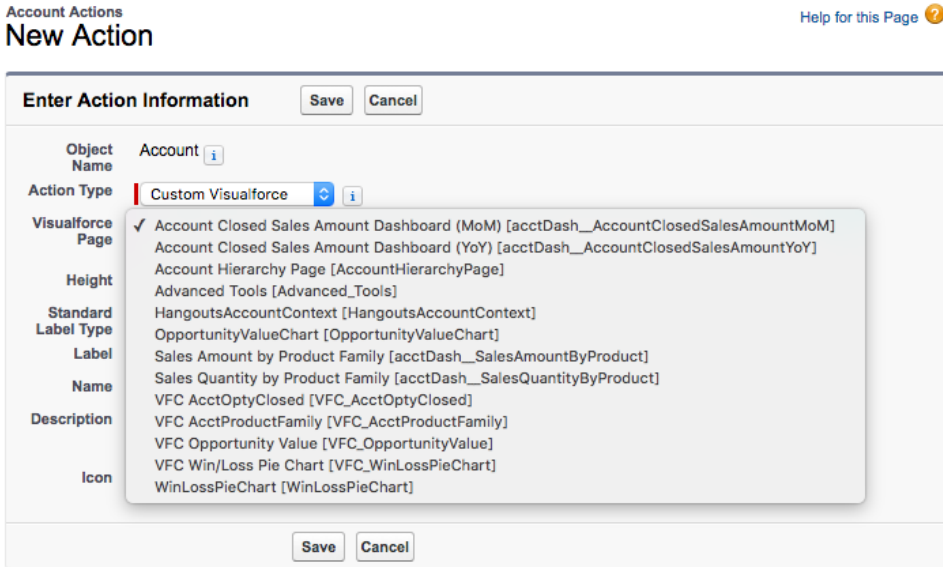
## Redirect to a Visualforce Page Based on Input Values

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.

## Chapter 22 You Have Buttons? We Have Alternatives

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.



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
<code>www.google.com</code>	
Relative Salesforce URL, View <code>/ {!Account . Id}</code>	Record home page opens in existing tab
Relative Salesforce URL, Edit <code>/ {!Account . Id} /e</code>	Edit overlay pops up on the existing page
Relative Salesforce URL, List <code>/001/o</code>	Object home page opens in existing tab
\$Action URL, View <code>{ !URLFOR (\$Action . Account . View , Account . Id ) }</code>	Record home page opens in existing tab
\$Action URL, Edit <code>{ !URLFOR (\$Action . Account . Edit , Account . Id ) }</code>	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 the Salesforce mobile app.

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.

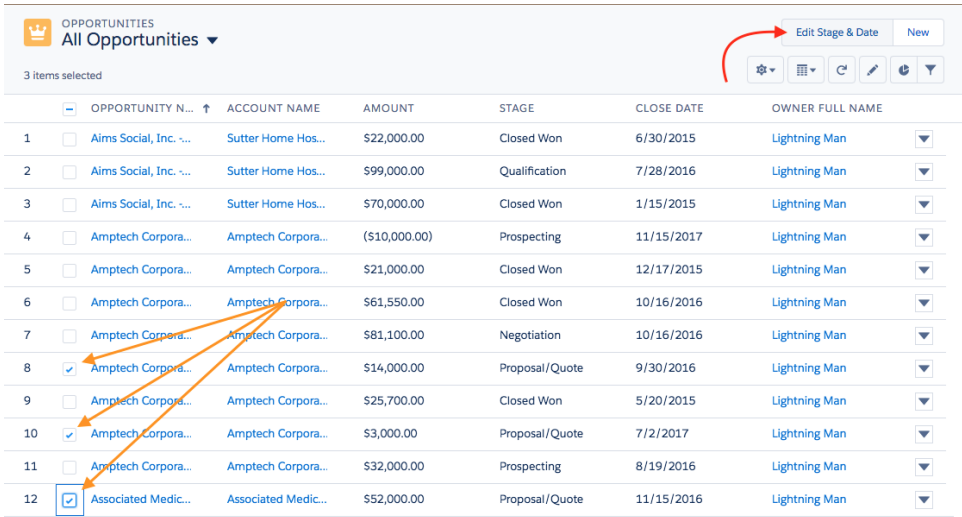
Here is sample code for editing the Stage and Close dates for multiple opportunities:

```
<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.



	OPPORTUNITY NAME	ACCOUNT NAME	AMOUNT	STAGE	CLOSE DATE	OWNER FULL NAME
1	<input type="checkbox"/> Aims Social, Inc. ...	Sutter Home Hos...	\$22,000.00	Closed Won	6/30/2015	Lightning Man
2	<input type="checkbox"/> Aims Social, Inc. ...	Sutter Home Hos...	\$99,000.00	Qualification	7/28/2016	Lightning Man
3	<input type="checkbox"/> Aims Social, Inc. ...	Sutter Home Hos...	\$70,000.00	Closed Won	1/15/2015	Lightning Man
4	<input type="checkbox"/> Amptech Corpora...	Amptech Corpora...	(\$10,000.00)	Prospecting	11/15/2017	Lightning Man
5	<input type="checkbox"/> Amptech Corpora...	Amptech Corpora...	\$21,000.00	Closed Won	12/17/2015	Lightning Man
6	<input type="checkbox"/> Amptech Corpora...	Amptech Corpora...	\$61,550.00	Closed Won	10/16/2016	Lightning Man
7	<input type="checkbox"/> Amptech Corpora...	Amptech Corpora...	\$81,100.00	Negotiation	10/16/2016	Lightning Man
8	<input checked="" type="checkbox"/> Amptech Corpora...	Amptech Corpora...	\$14,000.00	Proposal/Quote	9/30/2016	Lightning Man
9	<input type="checkbox"/> Amptech Corpora...	Amptech Corpora...	\$25,700.00	Closed Won	5/20/2015	Lightning Man
10	<input checked="" type="checkbox"/> Amptech Corpora...	Amptech Corpora...	\$3,000.00	Proposal/Quote	7/2/2017	Lightning Man
11	<input type="checkbox"/> Amptech Corpora...	Amptech Corpora...	\$32,000.00	Prospecting	8/19/2016	Lightning Man
12	<input checked="" type="checkbox"/> 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 23 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 the Salesforce mobile app and Lightning Experience.

You make a Lightning component invocable 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 Lightning 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

## Chapter 23 Lightning Actions: Smart, Fast, and Mobile

- 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.

### CreateUser.cmp

```
<aura:component
implements="force:lightningQuickActionWithoutHeader,force:hasRecordId">

    <aura:attribute name="user" type="Test_User__c" default="{ }"/>
    <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:attribute name="errorFromCreate" type="String"/>

    <!-- <aura:handler name="init" value="{!this}" action="{!c.init}"
/> -->

    <force:recordData aura:id="frd" mode="EDIT" layoutType="FULL"/>
```

```
<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}">
  <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

```

({
    /**
     * Auto generate the username from firstName on tab
     */
    updateNickname: function(component, event, helper) {
        // Update the nickname field when 'tab' is pressed
        if (event.getParams().keyCode == 9) {
            var nameInput = component.find("firstName");

```

## Chapter 23 Lightning Actions: Smart, Fast, and Mobile

```
        var nameValue = nameInput.get("v.value");
        var nickName = component.find("nickname");
            var today = new Date();
        nickName.set("v.value", nameValue + today.valueOf(today));

    }
},

/**
 * Capture the Inputs and invoke the helper.save with the input
params
 */
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) {
    // Create a user record, save it, and close the panel
    var userRecord = {apiName: 'Test_User__c', fields: {}};
    userRecord.fields.Name = {value: name};
    userRecord.fields.Password__c = {value: password};
    userRecord.fields.Email__c = {value: email};
    userRecord.fields.Nickname__c = {value: nickname};
    userRecord.fields.Case_Study__c = {value:
component.get("v.recordId")});
    // get the force:recordData and set the targetRecord
    component.find("frd").set('v.targetRecord', userRecord);
    // invoke saveRecord of force:recordData

component.find("frd").saveRecord($A.getCallback(function(response) {
  if (component.isValid() && response.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 (response.state == "ERROR") {
    console.log('There was a problem and the state is: '+
response.state);
  }
}));
  },

  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

```

## Chapter 23 Lightning Actions: Smart, Fast, and Mobile

```
wrong length (must be <= 15): " + value.length}}];
    } else if (value.search(/[0-9]+/) == -1) {
        inputCmp.set("v.errors", [{message: "The password must
contain at least one number."}]);
    } else if (value.search(/[a-zA-Z]+/) == -1) {
        inputCmp.set("v.errors", [{message: "The password must
contain at least one letter."}]);
    } 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);
    }
}
})
```

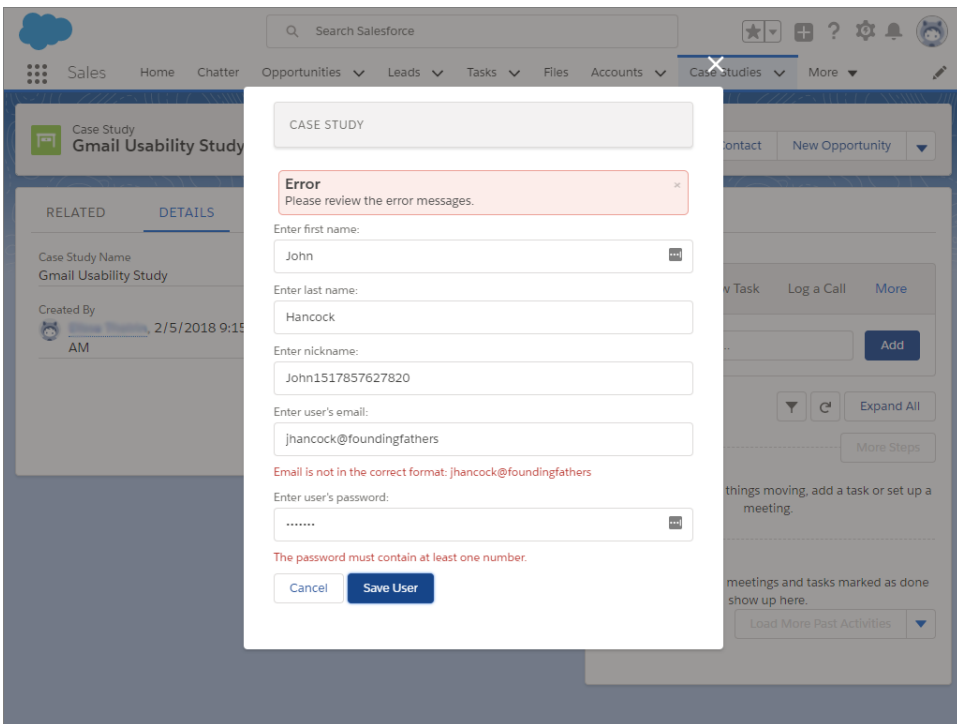
After creating the Lightning component, we assign it to an action. In the object management settings for Case Study, we go to Buttons, Links, and Actions, click **New Action**, then configure the action with these parameters.

Field	Value
Object Name	Case Study

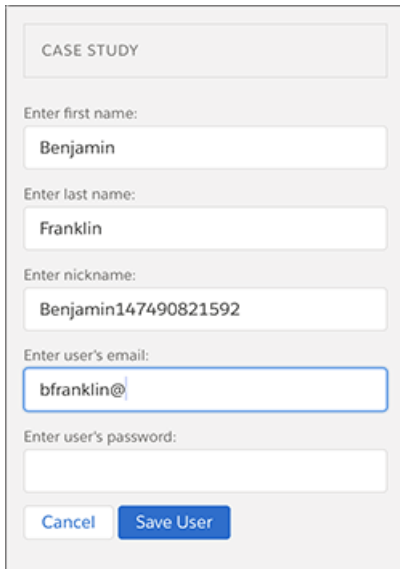


Field	Value
Action Type	Lightning Component
Lightning Component	c:CreateUser
Height	500px
Label	Create Test User
Name	CreateUser

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.



The great thing about this Lightning action is that it also works in the Salesforce mobile app.



A screenshot of a user creation form titled "CASE STUDY". The form contains several input fields and two buttons. The fields are labeled as follows: "Enter first name:" with the value "Benjamin"; "Enter last name:" with the value "Franklin"; "Enter nickname:" with the value "Benjamin147490821592"; "Enter user's email:" with the value "bfranklin@"; and "Enter user's password:" which is currently empty. At the bottom of the form are two buttons: "Cancel" and "Save User".

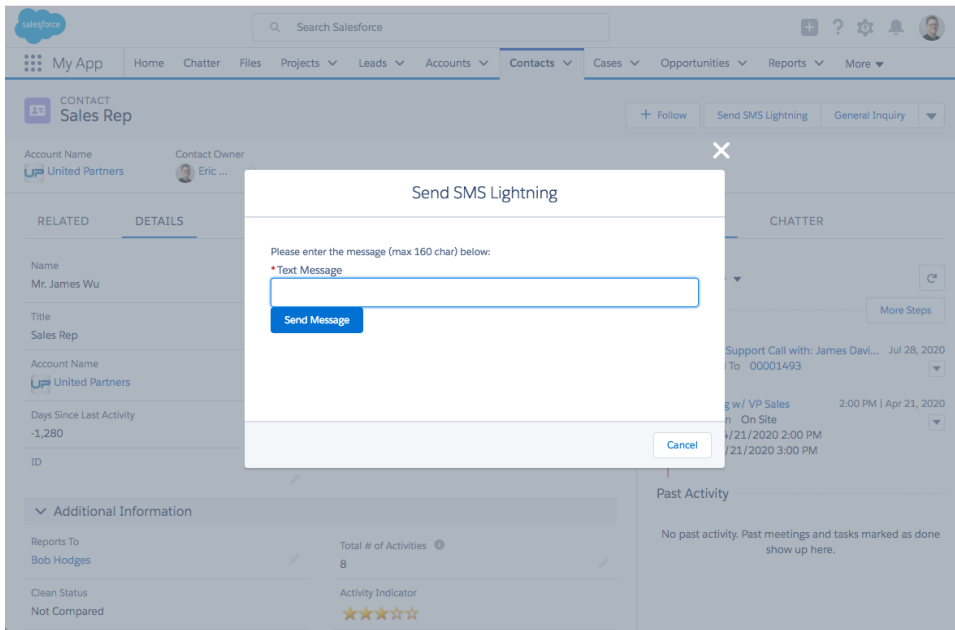
## Integrate Third-Party APIs

---

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.sendMessage");
    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) {

```

## Chapter 23 Lightning Actions: Smart, Fast, and Mobile

```
        mobNumber = getPhoneNumber(contactId);
    }

    try {
        TwilioRestClient client = TwilioAPI.getDefaultClient();

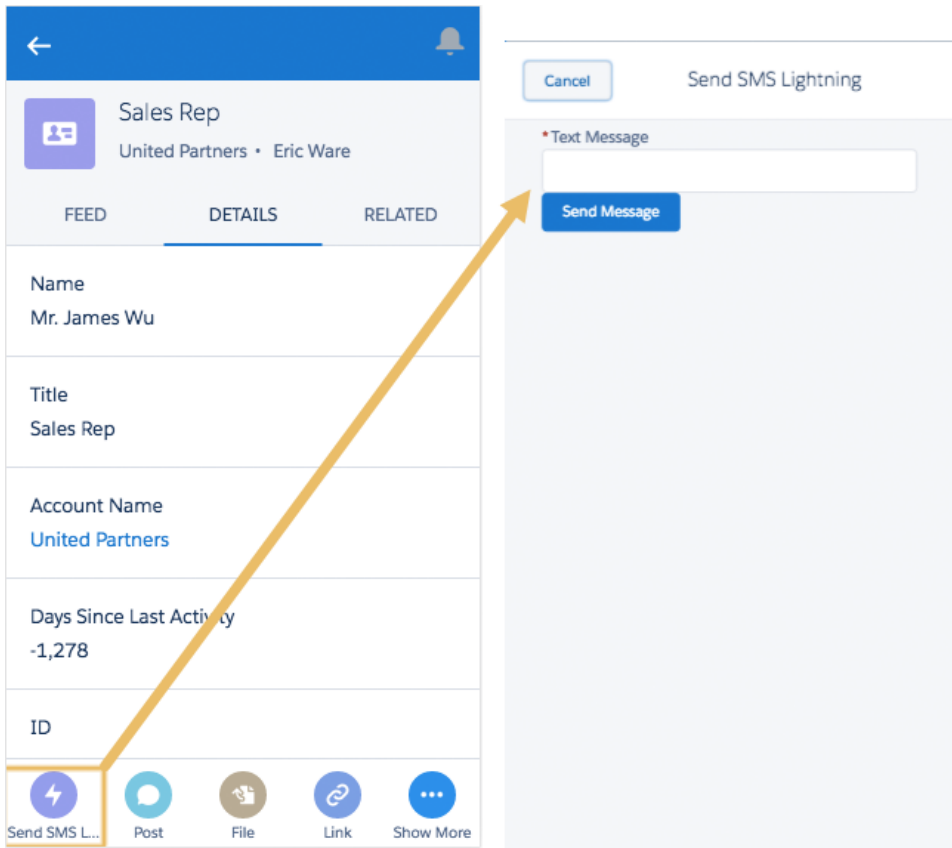
        Map<String,String> params = new Map<String,String> {
            'To' => mobNumber,
            'From' => '15555551234',
            'Body' => message
        };

        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 the Salesforce mobile app. 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.



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 24 Welcome to Your Lightning Experience Rollout

In this chapter ...

- [What It Takes to Go Live](#)
- [Educate Yourself About Lightning Experience](#)
- [Identify Stakeholders and an Executive Sponsor](#)
- [Revisit Your Processes](#)
- [Perform a Gap Analysis](#)
- [Improve Your Implementation with Salesforce Optimizer](#)
- [Get Hands-On with Lightning Experience](#)
- [Assess the Impact](#)
- [Educate Your Company About Lightning Experience](#)
- [How to Demonstrate Lightning Experience](#)
- [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 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 started with sales, Lightning Experience doesn't end there. Besides transforming Salesforce CRM, Lightning Experience extends 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.

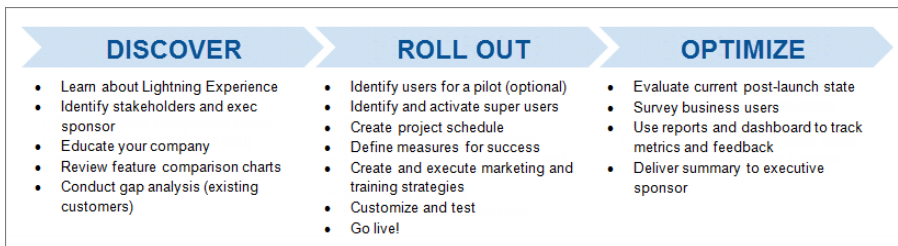
So let's get this show on the road!

## 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 find an executive sponsor and get stakeholders aligned on the value of switching to Lightning Experience, then you'll plan your rollout. You'll decide whether to start with a pilot for a set of users (like a preview) or go big and enable for all users at once (head straight to Broadway). And then you'll put on the show.



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 the new interface.

Let's begin there.

## Educate Yourself About Lightning Experience

---

Before you start defining your strategy to roll out Lightning Experience, spend some time learning about the ins and outs of the new interface. 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
Home	Start the day on the new, intelligent home page. Monitor performance to goal, and get news on key accounts. 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. Reps work their deals more efficiently using the handy composer to quickly log calls, create tasks, send emails, and more.
Path	Guide reps along the steps of a process, such as working an opportunity from a fresh lead to closing a deal. At each step, help your team succeed with specific guidance and resources.
Kanban	Visually review the status of your records for most objects. Drag records from one column to another. For opportunities, get alerts when an action is needed on a key deal.
Accounts and Contacts	Find information and see news at a glance from the new layout. Get the latest information about customers with integrated news articles, review upcoming and completed activities, and keep data clean with field-level duplicate matching.
Salesforce Einstein	Supercharge the productivity of your sales, service, and marketing teams with artificial intelligence. Using key predictions, intelligent recommendations, and timely automation, everyone can work smarter and faster.
Activity Timeline and Task List	<p>See open tasks, planned meetings, and accomplishments in the activity timeline on each opportunity, lead, account, and contact.</p> <p>Or use the task list to see details about specific tasks alongside the full list of open tasks, tasks due today, and closed tasks. Managers see tasks assigned to people below them in the role hierarchy.</p>
Notes	Take better notes with auto-save, rich text capabilities, inline images, and versioning. Relate a note to multiple records and share notes with teammates or Chatter groups.
Integrated Email	Relate email messages to multiple contacts, leads, and users, and to a single opportunity, campaign, case, account, or person account. Send email through Gmail or Office 365 accounts with your Salesforce email. And see the email messages you've sent in your Gmail or Office 365 Sent Items folder for seamless integration.

## Chapter 24 Welcome to Your Lightning Experience Rollout

Feature	Why It's Cool
Composer Windows	Breeze through those minor but important activities that inevitably crop up during the day without switching context. Quickly log calls, create tasks and events, and jot down notes, all from windows that open on the current page.
Lightning Dialer	Use phone features without ever leaving Salesforce. Make and receive calls, add call notes, and log call information with ease.
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. Scroll through multi-page documents, preview animated GIFs, give full-screen presentations, and access file actions.
Live Feeds	See real-time posts in group feeds and real-time comments in all feeds without refreshing the page. Even see who's commenting while they're typing.
Streams	Easily track related information in one place when you combine up to 25 feeds into one super-feed.
Post Pinning	Pin as many as three posts to the top of group and topics feeds. Highlight high-impact posts that otherwise get buried in a busy feed.
Reports and Dashboards	Get more from your data. The report run page is easy to read and filter. Animated, interactive charts bring your company'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 quickly 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.
Themes and Branding	Rally your team around your brand and important initiatives, such as a new product launch, with a set of custom brand images and colors.
Personalized Navigation	Supercharge your productivity by personalizing the Lightning Experience navigation bar to suit the way you work. Reorder and add navigation items, such as a frequently used dashboard or list, to the navigation bar.
Favorites	Create shortcuts to frequently accessed pages, including records, lists, and groups.

<b>Feature</b>	<b>Why It's Cool</b>
Search	Find records faster with improved global search, which includes top results and recent records.
Lightning Apps	Easily switch between collections of related objects, Visualforce and Lightning pages, and utilities such as Lightning Dialer. Customize the branding of each app with logos and color.
Lightning Console Apps	Navigate and work more productively with optimized workflows and a three-column layout. View and edit multiple records at the same time and quickly access utilities such as Lightning Dialer and Open CTI.
Setup Tree and Object Manager	Navigate setup pages with a simplified organization and a cleaner look and feel. Use the Object Manager, which combines all standard and custom objects into a single list that's easier to search.
Service Setup Flows	Get up and running quickly using the Lightning Service Console, which includes service-specific setup flows. These flows walk you through setting up Email-to-Case, integrating with Twitter and Facebook, creating a Lightning Community, and enabling Lightning Knowledge, Omni-Channel, and Live Agent.
Lightning Flow for Service	Guide your users through multi-step processes and associate flows with records using Salesforce automation tools. When opening records with associated flows, the flows launch either as a subtab in console apps or as a pop-up window in standard navigation apps.

## 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 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.

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”).

When you've identified an executive sponsor and your stakeholders, involve them in your project early and often, sharing everything you've learned about Lightning Experience with them.

With your new project team, and a clear understanding of the benefits you'll get from Lightning Experience, you're ready to evaluate your business and technical readiness for Lightning Experience. What are the opportunities to streamline your current business process by harnessing Lightning Experience features? How well will your org's features and customizations work as-is in the new interface? How much work is actually required before you can make the switch? Time to find out!

## Revisit Your Processes

---

As you and your stakeholders explore the features and benefits of Lightning Experience, there will likely be questions that sound like this: “That's cool, but can it do this thing we do today? What about that other thing we do—can it do that?”

It's natural to be thinking about how you can take your existing business process and make it work in the new user experience. 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. More than just a new interface, Lightning Experience re-imagines how you manage your business. As trusted advisor, you can help your users see that updating their process to adopt new features and paradigms in Lightning Experience can actually make them more productive.

And if you've been a Salesforce customer for a long time, no doubt your business has evolved and changed. This is a great opportunity to re-evaluate what your users need and align your Lightning Experience implementation with those needs.

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 an existing process.
- Walk key members of your sales teams and super users through Lightning Experience 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.

## Perform a Gap Analysis

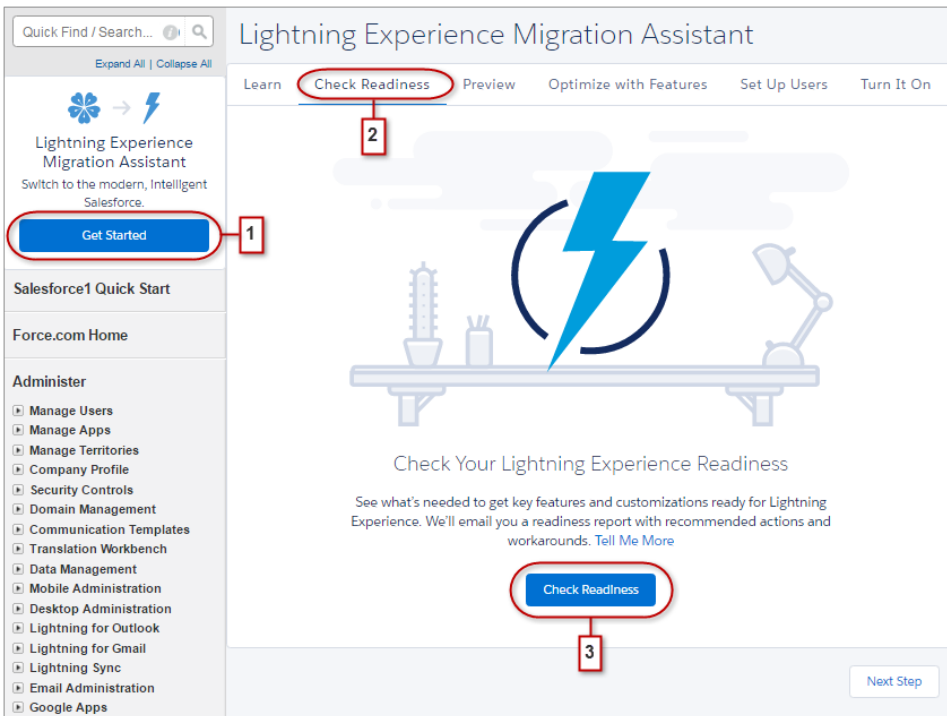
---

Performing a gap analysis is where you get into the nitty gritty details of how your current Salesforce implementation will 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 changes or workarounds you might need to put in place. You may even be delighted to learn that you're ready for Lightning Experience now!

Luckily, we have the perfect tool to make light work of this analysis. Meet the Lightning Experience Readiness Check!

Access the Readiness Check from the Lightning Experience Migration Assistant. From Setup in Salesforce Classic, click **Get Started** in the Migration Assistant tile at the top of the menu (1). On the Check Readiness tab in the Migration Assistant (2), click **Check Readiness** (3).

## Chapter 24 Welcome to Your Lightning Experience Rollout



After you kick off the Readiness Check, Salesforce generates a PDF Readiness Report and emails it to you. Salesforce also saves the report to Salesforce Files.

Review your personalized Readiness Report for advice and recommendations about how to get ready for Lightning Experience. Use the findings in your report to identify the work that should happen before you launch a pilot, updates that can happen in a later phase of your rollout, and the users who are good candidates for your pilot group.

Curious what exactly is covered in the report? Check out [What Does the Lightning Experience Readiness Check Evaluate?](#) in the Salesforce Help.

If you like, you can set up a free 30-minute consultation with a Lightning expert to review your report and ask questions about your next steps.

The Readiness Check doesn't analyze all Salesforce features and customizations. To help you evaluate things not covered by the Readiness Check, we've included a sample gap analysis checklist in the [enablement pack](#). It highlights many of the areas you should investigate when conducting your gap analysis.



## Improve Your Implementation with Salesforce Optimizer

---

Think about the things that happen when you move to a new house. You plan for improvements at the new place. You donate stuff you don't use anymore instead of packing it up. Maybe you get a new sofa that fits better in the new space. It's just like that when planning your org's move to Lightning Experience! This is a great opportunity to evaluate the effectiveness of your org's features and customizations instead of simply moving everything as-is to the new interface. To get started without any heavy lifting required, take a gander at Salesforce Optimizer. The tool identifies things you can simplify, improve, or leave behind when transitioning to Lightning Experience—so you know where to concentrate your time and resources.

Optimizer generates a report that's specific to your implementation, uploads it as a file, and emails it as an attachment to the admin who runs the report. The report provides recommendations to improve feature use, clean up customizations, and reduce complexity.

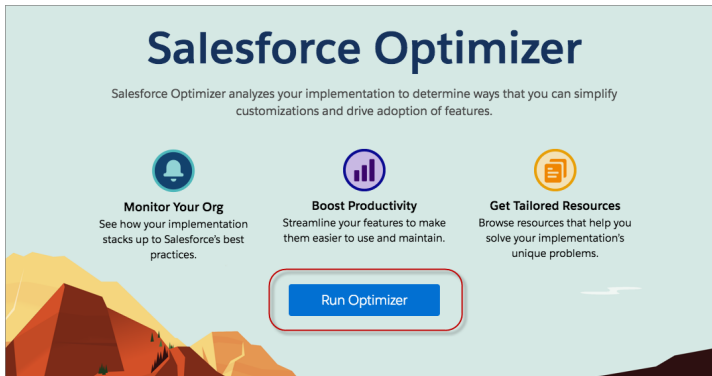
Some of the features that Optimizer evaluates include Apex triggers, fields, profiles, record types, and validation rules. For the complete skinny, take a look at [Which Features Does Optimizer Evaluate?](#) in the Salesforce Help.

The Optimizer report provides recommendations that can make your move to Lightning Experience more efficient. Here are just a few examples.

- Page layouts are displayed differently in Lightning Experience than they are in Salesforce Classic. The more page layouts you have to manage, the more work you have to do to make sure they all display properly in the new interface. The Optimizer report identifies page layouts that aren't being used in your implementation, so you can delete them before you transition to Lightning Experience.
- Hard-coded links in your implementation will likely break in Lightning Experience. The Optimizer report tracks down any hard-coded URLs that are lurking in your org and flags them for you. Replace them with My Domain links, which won't break when you turn on Lightning Experience.
- Older web browsers aren't supported by Lightning Experience. The Optimizer report lists users who access Salesforce from unsupported browsers so you can help them update to a supported version.
- Lightning Experience includes many productivity-enhancing features that aren't available in Salesforce Classic. The Optimizer report recommends Lightning Experience-only features that your users may benefit from, such as the Lightning Sales or Service Consoles.

Let's start optimizing!

1. From Setup, enter *optimizer* in the Quick Find box, then select **Optimizer**.
2. Click **Launch > Allow**.



Then kick your feet up and relax while Optimizer gets to work generating your report.

Depending on the size of your implementation, the Optimizer report takes from a few minutes to several hours to run. When it's ready, you get a copy in email and a copy is saved to Salesforce Files.

## Get Hands-On with Lightning Experience

---

One of the best ways to evaluate how your org's features and customizations work in Lightning Experience is to get some hands-on experience in the new interface. If you'd like to try out Lightning Experience before you enable it in your production org, you've got options.

- Use the Preview tool that's available from the Lightning Experience Migration Assistant. Preview 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.

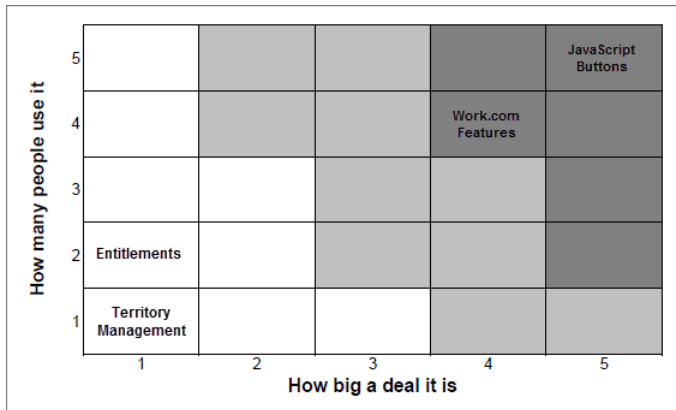
## Assess the Impact

---

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 Territory Management and have limited use of Entitlements, 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.



## Educate Your Company About Lightning Experience

When you’re finishing assessing the impact of each gap, you’re 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, 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.

One of the most important questions your executive sponsor and stakeholders will have is, “How will Lightning Experience help my team sell more?” So start your presentation by showcasing the benefits of Lightning Experience. To help you share Lightning Experience with your company, get the presentation deck in this [enablement pack](#), which highlights features and benefits.

If you’re an existing Salesforce customer, explain if any features that are important to your company aren’t available in Lightning Experience today. Your stakeholders can use this information to make an educated decision on when to enable the new interface for your different groups of users.

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!

## 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 25 All Systems Go

### In this chapter ...

- [Where to Start](#)
- [Plan Your Rollout](#)
- [Decide How to Structure Your Rollout](#)
- [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?

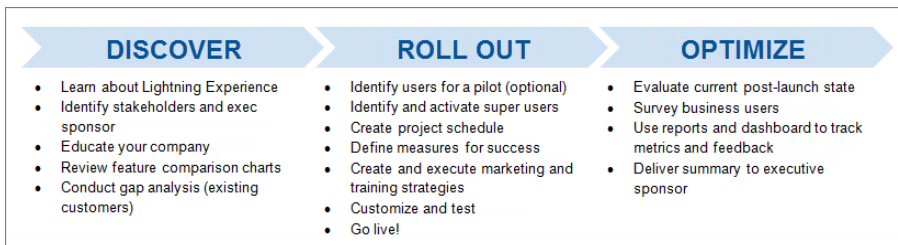
Doing all of that takes time and coordination. There are a number of methodologies for project management, too many for us to cover here. Each one has different strategies for delivering projects. Pick whichever method you like. Ultimately, the important part isn't the way you do it; it's what you do. 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 phases. You already learned about the "Discover" steps in the last unit. In this chapter, we'll look at the "Roll Out" phase.



Let's dive in.

## Decide How to Structure Your Rollout

---


There isn't a single right way to roll out Lightning Experience. There are several variables to consider when deciding on a strategy—like the complexity of your implementation and the size of your Salesforce user

base. But in most cases, you'll take one of these approaches: transition your org gradually in phases or go all-in, switching your entire org at the same time.

To pick the best strategy for your org, let's look at the benefits and considerations for both approaches.

## Phased Rollout

Unless your org is small, it can be advantageous to transition your users to Lightning Experience over the course of several phases. This approach allows you to tackle your business and technical requirements incrementally, and iterate on lessons learned. For each phase, pick teams or groups of users whose business needs are met by the current state of your Lightning Experience implementation. After launching a phase, collect feedback. Then fine-tune things and kick off the next phase.

 **Note:** If you're using Group Edition, this approach isn't an option for you. Lightning Experience is "all or nothing"—all users get access when Lightning Experience is enabled.

Benefits	
Run a pilot to learn about, and refine, your implementation	Ensure that users are productive and enthusiastic about adopting Lightning Experience by starting small with a pilot program. Using the first phase to conduct user acceptance testing allows you to gather feedback from pilot users and solidify your implementation before going all-in with the next phase.
Break up the work and avoid technical debt	For each phase, focus your configuration and development efforts on just what's necessary to make the users in that phase productive. Chunking up the work allows you to make continual progress on your implementation and avoid the pitfalls of falling behind on new features and enhancements as they're released.
Get fine-grained control over which users get access to Lightning Experience	Use permission sets for a flexible way to roll out Lightning Experience to a particular team or group of users. Rather than updating your custom profiles, create a permission set that includes the Lightning Experience User permission, then assign the permission set to the users you've identified for each phase.

<b>Considerations</b>	
Keep teams together	<p>Put people who work closely together in the same phase. You want people who work together seeing the same screens. If people collaborate often, they should have the same user experience, including team leaders.</p> <p>If you have standard profile users who shouldn't get access to Lightning Experience in earlier phases, plan to move them to custom profiles that don't include the Lightning Experience User permission.</p>
Be mindful of gaps	<p>If you have teams that are heavily impacted by issues that you identified in your gap analysis, avoid including those teams in a phase until the issues are addressed. Even if the number of gaps is small, if the impact is large, consider carefully if they'll be productive working in Lightning Experience.</p>
More training and other change management is required	<p>You'll need to invest more time in change management activities. For example, you'll need to repeat communication and drip campaigns for each phase. And depending on how far apart you launch each phase, you may also need to run user trainings multiple times.</p> <p>If your user base is split across Lightning Experience and Salesforce Classic, you may need to maintain training materials for both interfaces. Though you can avoid this by enabling all new users for Lightning Experience and eliminating their ability to switch to Salesforce Classic.</p>

## Org-Wide Rollout

With this approach, you enable Lightning Experience for all users in your org at the same time. On the surface, this may seem like the easiest, most straightforward way to structure your rollout. If your org configuration is simple and you don't have many Salesforce users, this method may make the most sense for you. But it also requires that you're highly confident that everyone gets what they need in the new



interface and that everyone is prepared for the transition. And obviously, the need for advanced preparation has implications for how long it will take to get ready to flip the switch.

<b>Benefits</b>	
Simpler training and change management landscape	With all your users moving to Lightning Experience at the same time, you can run a single campaign to get everyone on board. And you only need to maintain training materials for one interface.
Simplified feature and program enablement	You don't need to evaluate new features, customization work, integrations, or partner applications for multiple interfaces. All your users get the benefits of new features and technology as they're released.
<b>Considerations</b>	
Delays time to launch	Getting Lightning Experience ready for everyone in your org means you're front-loading all the preparation work, including implementing all required business processes and addressing all technical issues and feature gaps that you identified. In other words, it can be quite time consuming to get ready for this approach.
Less feedback increases the chance of missing something important	Waiting to go all-in with everyone means that you don't get the advantage of early feedback on how things are working. If you inadvertently miss an important use case or customization, everyone is impacted instead of a smaller subset of users. If users don't get what they need out of the gate, they lose their enthusiasm. Ultimately, adoption rates could suffer.
Watch out for technical debt	We're continually improving and enhancing Lightning Experience. The longer it takes to move your org over, the more features and opportunities for improved productivity you'll get behind on.

## Decide Which Users Get Lightning Experience


If you're taking a phased approach to your rollout, there are things to consider when deciding who moves to Lightning Experience in each phase. And for each phase, you have options to control who is immediately switched to the new interface and who can and can't switch between Lightning Experience and Salesforce Classic. Get all the details from [Set Up Users for Lightning Experience](#) on page 43.

Whether you're deciding on the project schedule for a phased rollout or an org-wide rollout, picking your launch date is influenced by the users involved and what they'll get from Lightning Experience when they're enabled. You may be inclined to hold off on launching to your users until everything is fully implemented. Or maybe you're fretting about grumpy users—you know, users who have a lot going on and no time to learn a new way of working. These are legitimate considerations but they shouldn't be a reason to delay getting started. Saying hello to Lightning Experience doesn't mean saying goodbye to Salesforce Classic! Unless you set it up otherwise, your users can switch themselves between both interfaces as needed. So they can easily pop back to Salesforce Classic if something they need isn't available yet in Lightning Experience.

## 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

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.

Chatter > People > Joan Moreno



## 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.

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 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 <i>Lightning Experience Guide</i>	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 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 and 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:

<b>If the customized feature does this:</b>	<b>Consider using this instead:</b>
Creates a related record	Process Builder
Updates an existing record	Actions
Creates related records and updates existing records, with complex logic	Process Builder and Cloud Flow Designer
Launches a screen for user input	Cloud Flow Designer
Sends an email or creates a task	Process Builder
Launches a time-triggered process	Process Builder

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 26 Go Live

### In this chapter ...

- [Drive Adoption by Optimizing Your Implementation](#)
- [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 **Get Started** in the Migration Assistant tile.

The Migration Assistant is your guide to enabling Lightning Experience. Keep in mind there are other features that we recommend you enable before turning on Lightning Experience. Key features are covered by the Migration Assistant. You can also review [Recommended Features for Lightning Experience](#) on page 42 in the Salesforce Help for more details.

## Drive Adoption by Optimizing Your Implementation

---

Encores and repeat performances... Just like a Broadway production, your Lightning Experience rollout keeps going after your launch. Whether you initially go live with a pilot group or you enable your entire org at once, you want to make sure that your users stay enthusiastic and committed to working in Lightning Experience.

This is the time to rinse and repeat. Or rather, optimize your rollout by measuring how things are going and iterating on all the good work you've done so far.

Monitor how your success metrics are looking. For example, collect feedback on how users are liking the new experience. And measure for things like productivity gains, business success, and adoption rates—including how many users are switching back to Salesforce Classic and why.

With this data in hand, you'll have a solid idea of what your next steps should be. You'll know which new use cases are priorities for fine-tuning your implementation, which features on the Lightning Experience roadmap are important, and when you'll be ready to add another round of users to Lightning Experience.

## 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:

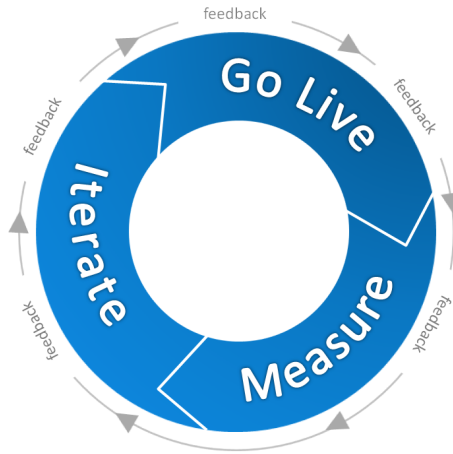
Open-ended questions	Survey questions
How do you use Salesforce?	How long have you been using Salesforce?
What do you like best about Salesforce?	Rate your overall productivity using Salesforce.
What would you change about Salesforce?	Rate your overall satisfaction with Salesforce.
What is most frustrating about Salesforce?	I have the tools I need to do my job: Yes or No.
What information do you need that you can't find?	It's easy to work in Salesforce: Yes or No.

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

##### Project summary

- Start date: September 1
- Launch date: October 15
- Rollout to 35 sales reps in the West Geo

##### Measures for success

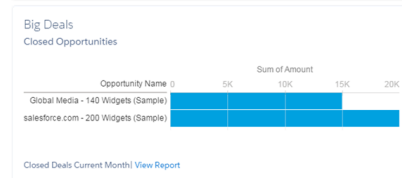
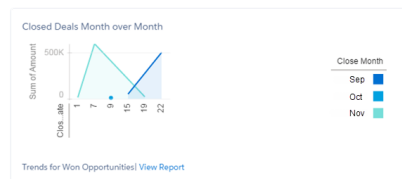
- 5% increase in converted leads
- 5% increase in employee satisfaction

##### Results

- 8% increase in converted leads
- 10% increase in employee satisfaction

##### Anecdotes from survey

- "I love the Opportunity Kanban and filters. It's so easy."
- "The new home page is great. I want even more charts on it."



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 27 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 Lightning Platform 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 Lightning Platform 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 Lightning Platform 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 Lightning Platform 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.

## Chapter 27 User Interface Development Considerations

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
<b>I'm Developing for Lightning Experience</b>	<p>We highly recommend Lightning Components. Lightning Experience was <i>built</i> with Lightning Components, and the two fit together like hand and glove.</p> <p>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.</p> <p>But you won't find a better tool for Lightning Experience than working in its native language, Lightning Components.</p>
<b>I'm Developing for the Salesforce Mobile Application</b>	<p>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 <i>specifically</i> to handle this context.</p> <p>Both Visualforce and Lightning Components use similar tag-based markup. For example, Visualforce markup for an input field is <code>&lt;apex:inputText&gt;</code> while for Lightning Components it's <code>&lt;lightning:input&gt;</code>.</p> <p>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.</p> <p>With a few exceptions, Lightning Components are better for Salesforce mobile development. Some cases call for Visualforce as a JavaScript application. See the Lightning Components Developer's Guide for more information.</p>



Job	Recommendation
<b>I'm Building a Page-Centric Experience with Limited Client-Side Logic</b>	Use Visualforce pages to ensure development velocity and manageability.
<b>I'm Building an Interactive Experience with JavaScript to Meet User Experience Requirements</b>	Use Lightning Components but refer to limitations documentation first.
<b>I'm Committed to a JavaScript Framework Such as AngularJS or React</b>	Use a Visualforce page as a container for your third-party framework, such as AngularJS or React, and your application.
<b>I'm Enabling Non-Developers to Build Apps by Assembling Standard or Custom Components</b>	Use Lightning Components to create custom components that can be used in the Lightning App Builder.
<b>I'm Building an Interactive Experience with JavaScript and I Need a Third-Party Framework</b>	Use a Visualforce page as a container for your third-party framework, such as AngularJS or React, and your application.
<b>I'm Adding User Interface Elements</b>	For example, say you want to add a tab to a record home. This task is a simple drag-and-drop in the Lightning App Builder. Use Lightning Components to create custom user interface elements.
<b>I'm Building a Community for Customers</b>	Use Community Builder to create a Lightning-based community leveraging Lightning Components.
<b>I'm Building a Community for Partners</b>	Use Community Builder to create a Lightning-based community leveraging Lightning Components.
<b>I'm Exposing a Public-Facing Unauthenticated Website</b>	Use Community Builder to create a Lightning-based site leveraging Lightning Components.
<b>I'm Rendering Pages as PDFs in My Application</b>	Use Visualforce. Lightning Components don't support rendering as PDF output yet.
<b>I'm Adding to an Existing Project with Lots of Visualforce Pages</b>	Continue to use Visualforce. Consider opportunistically moving elements to Lightning Components using Lightning Components for Visualforce.

<b>Job</b>	<b>Recommendation</b>
<b>I'm Committed to Investing in the Latest Technology</b>	You're right there with us! Dive in to Lightning Components. Start with the Lightning Components Basics Trailhead module.
<b>I'm Starting a Brand New Project</b>	Use Lightning Components. If you're not familiar with them, there's no better time than now to learn!
<b>I'm Building a Flow to Guide Users Through a Business Process</b>	Use Lightning Components to customize the functionality and look-and-feel of the flow's screens.

## 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.

<b>Role</b>	<b>Recommendation</b>
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 the Lightning App Builder.
Point-and-click admins	Use the 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 28 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 [Use 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 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.

## Chapter 28 Visualforce and Lightning Experience

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 most standard and all custom objects 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, along with some elements of object tab, like recent items. Other elements, 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 `sidebar` 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 29 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 covered some 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 need to do things like render PDFs from a page. Visualforce hasn't gone away, and remains a foundational part of developing on the Salesforce platform.

The Lightning Components framework is, relatively speaking, the new kid on the block—but this new kid has skills. While there are a few specific limitations of Lightning components, for the most part it's ready to go. There are many situations where you should consider making the switch to developing with Lightning components. Salesforce 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.

Let's make a list! Where exactly can you use Lightning components? You have many 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.

### **Salesforce App**

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 Lightning components and run it wherever your users are.

## Chapter 29 Lightning Components in Lightning Experience

### Screen Flows

If you haven't taken advantage of screen flows yet, you're missing out on a powerful way to guide your users through a business process. Customize the look-and-feel and functionality of your flow's screens by adding Lightning components to them.

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 30 Salesforce Partners, Packaging, and AppExchange

In this chapter ...

- [Partner Tools in Lightning Experience](#)
- [Packaging Apps in Lightning Experience](#)
- [AppExchange and Lightning Experience](#)

If you're a Salesforce AppExchange partner (ISV), you're probably wondering how Lightning Experience affects your development and release processes. Do your existing products still work? Do you have to make major changes to existing procedures? What about future releases?

These are valid concerns, but we have good news. Many of the tools that you use to build, test, and distribute your products are available in Lightning Experience. We also have resources to guide you as update your products to be Lightning Ready.

## Partner 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
<ul style="list-style-type: none"><li>• Environment Hub</li><li>• License Management App (LMA)</li><li>• Checkout Management App (CMA)</li></ul>	Both Salesforce Classic and Lightning Experience.
<ul style="list-style-type: none"><li>• Trialforce</li><li>• Channel Order App (COA)</li><li>• Usage Metrics Visualization App</li></ul>	Salesforce Classic only.

This means you can go to Lightning Experience if you want to:

- Create orgs for development, testing, and demos
- Manage all your orgs from a single location
- License and support apps published on the AppExchange
- View and report on subscription data for products sold using AppExchange Checkout

To provision trial orgs or manage orders, you should stay in Salesforce Classic. As Lightning Experience continues to mature, more tools will become available.

## Packaging Apps in Lightning Experience

---

When you're ready to distribute an app, Lightning component, or other product, 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 offer a product on AppExchange, you probably have questions about getting Lightning Ready. We consider products Lightning Ready when all end-user use cases work as expected in Lightning Experience.

So, how long do you have to get Lightning Ready? The answer depends on the type of product and your distribution agreement with Salesforce. New products sold using AppExchange Checkout and new free products must be Lightning Ready before their security review. Other agreement types may require Lightning Experience readiness depending on when you signed the agreement. For existing products, we encourage you to get Lightning Ready as soon as possible. For more information and resources, go to the Lightning Ready page on the Salesforce Partner Community.

If you're an AppExchange customer, look for the Lightning Ready certification on a product's listing to verify that it's compatible with Lightning Experience. You can use products that aren't certified as Lightning Ready in Lightning Experience, but there's no guarantee that they'll work as expected. It's best to use them in Salesforce Classic.



## CHAPTER 31 Understand 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](#)
- [Salesforce Mobile 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

---

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, the

Salesforce app, and everywhere else they're supported—with the added bonus that you can integrate Canvas apps in Lightning Components!

## Salesforce Mobile for Lightning Experience

---

Lightning Experience and Salesforce mobile 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 `sfForce.one` JavaScript object. In the past, it was used as an event mechanism for navigation in Salesforce mobile development. Now, you can also use it for navigation in Lightning Experience. See the Salesforce 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 32 Use 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

## **Chapter 32** Use Visualforce in Lightning Experience

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.




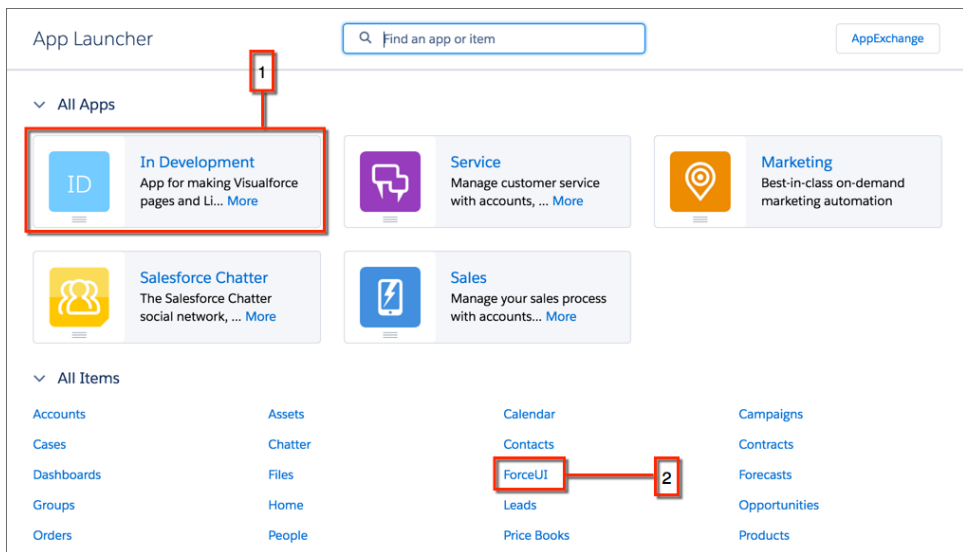
## Where You Can Use Visualforce in Lightning Experience

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

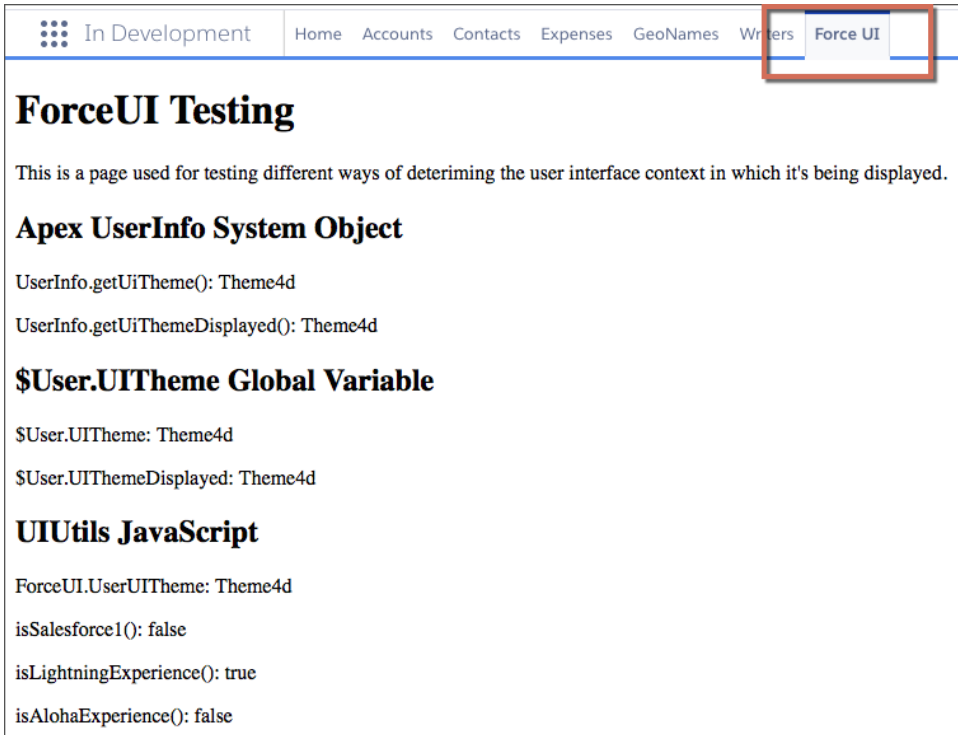
Your Visualforce apps and custom tabs are all available from the App Launcher, which you reach by clicking  in the header.



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.



The screenshot shows a Visualforce page with a navigation bar at the top. The navigation bar includes a menu icon, the text 'In Development', and several tabs: 'Home', 'Accounts', 'Contacts', 'Expenses', 'GeoNames', 'Writers', and 'Force UI'. The 'Force UI' tab is highlighted with a red rectangular box. Below the navigation bar, the page content is as follows:

## ForceUI Testing

This is a page used for testing different ways of determining 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

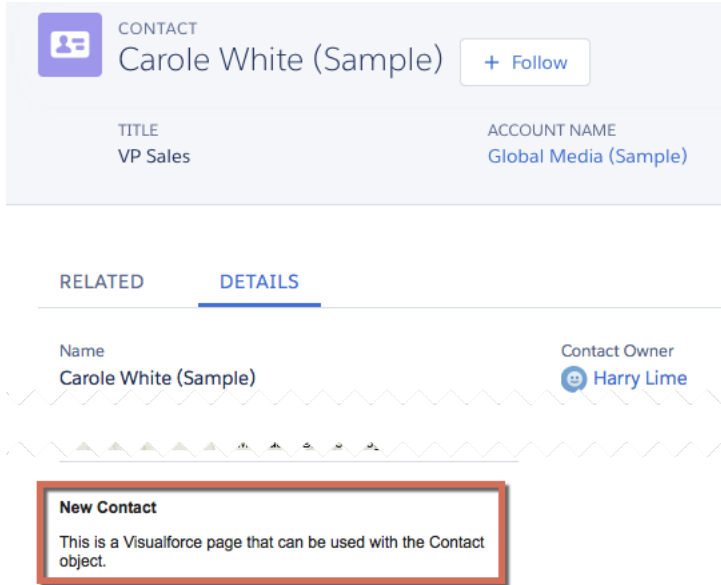
### UIUtils JavaScript

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.

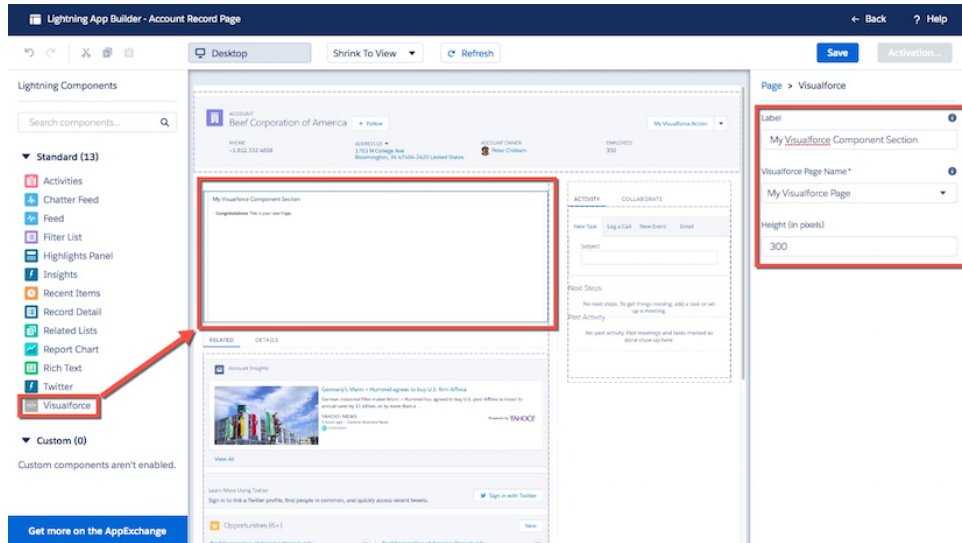



The screenshot displays a Salesforce Contact record for "Carole White (Sample)". The record header includes a contact icon, the name "Carole White (Sample)", and a "+ Follow" button. Below the header, the "TITLE" is "VP Sales" and the "ACCOUNT NAME" is "Global Media (Sample)". The "DETAILS" tab is selected, showing the "Name" as "Carole White (Sample)" and the "Contact Owner" as "Harry Lime". A red-bordered box highlights a "New Contact" section containing the text: "This is a Visualforce page that can be used with the Contact object."

## Chapter 32 Use Visualforce in Lightning Experience

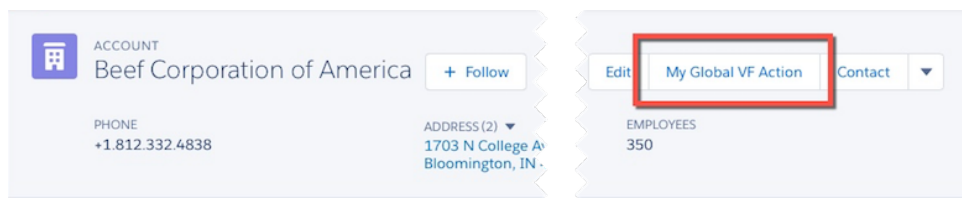
### Example: Add a Visualforce Page as a Component in the Lightning App Builder

When you create a custom app page in the Lightning App Builder, you can add a Visualforce page to the page by using the Visualforce component.

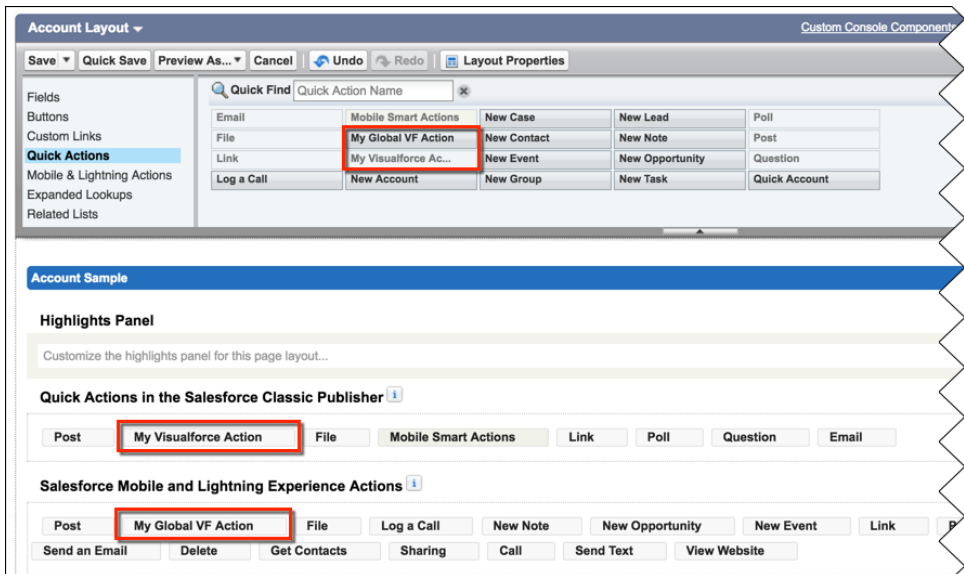


 **Note:** You must enable "Available for Lightning Experience, Lightning Communities, and the mobile app" for a Visualforce page to make it available in the Lightning App Builder.

### Example: Launch a Visualforce Page as a Quick Action

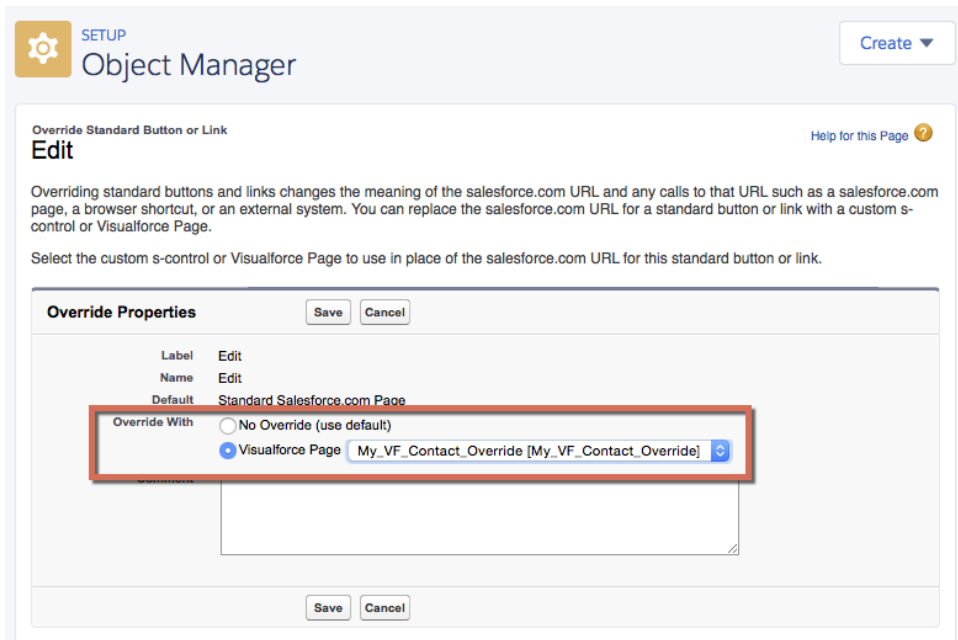


Although their placement in the Lightning Experience user interface is quite different from Salesforce Classic, the process of adding quick actions is much the same. Add them to the appropriate publisher area on the object's page layout.



**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!



**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 33 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 Salesforce mobile 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 the Salesforce app, 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 can't 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 `/lightning`. 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 are a variety of ways to do that.

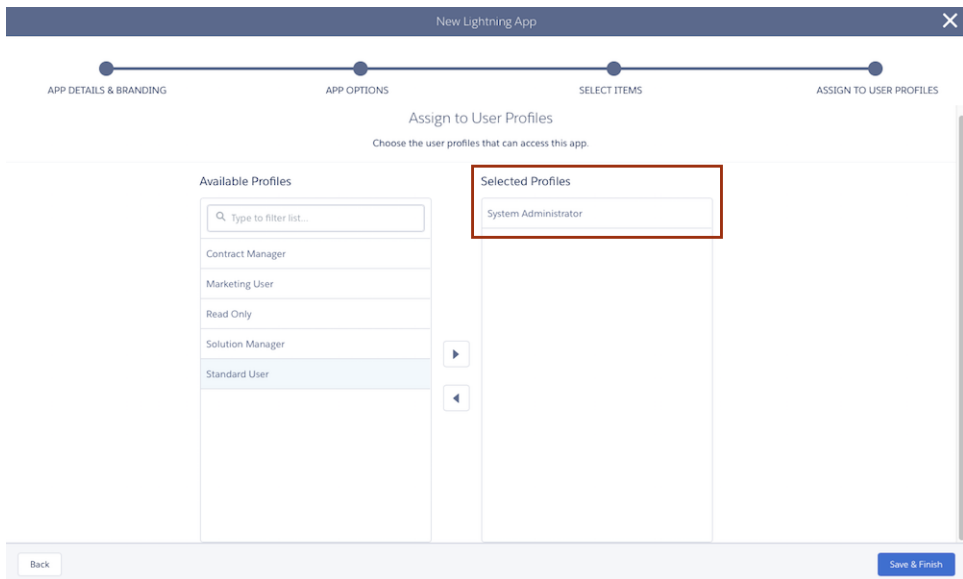
The simplest way to get to a specific Visualforce page is to create a tab for it, and then navigate to that tab via the All Items section in the App Launcher. A more long-term approach would be to create an "In Development" app, and add your Visualforce tabs to it as you work on them, and move or remove 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.

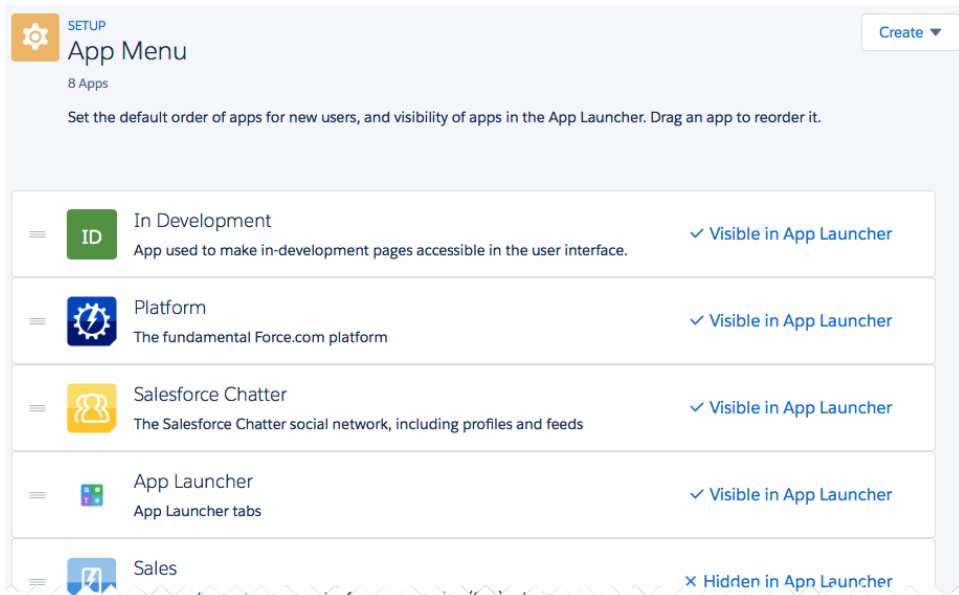
Consider restricting your app to only System Administrators, or a profile you've created for developers in your organization.



You don't need your users to see your pages before they're added to their permanent place in the user interface.

## Chapter 33 Developing Visualforce Pages for Lightning Experience

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 that 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.



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	<input type="checkbox"/> Include Tab
Platform	<input type="checkbox"/>
Sales	<input type="checkbox"/>
Salesforce Chatter	<input type="checkbox"/>
Marketing	<input type="checkbox"/>
Call Center	<input type="checkbox"/>
Profile (Self)	<input type="checkbox"/>
Profile (Others)	<input type="checkbox"/>
Community	<input type="checkbox"/>
App Launcher	<input type="checkbox"/>
In Development	<input checked="" type="checkbox"/>

Append tab to users' existing personal customizations


- 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();
```

## Chapter 33 Developing Visualforce Pages for Lightning Experience

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 the Salesforce app, 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 Salesforce mobile 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

### Salesforce Mobile Review Environment

This environment is for checking your page's design and behavior in the Salesforce app.

- **Device:** iOS or Android phone or tablet
- **Browser:** Salesforce 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 the Salesforce app.

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.

## Chapter 33 Developing Visualforce Pages for Lightning Experience

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.

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 Salesforce app, which embeds its own.
- Each different supported user interface context (Lightning Experience, Salesforce Classic, and the Salesforce app).

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 Salesforce 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 34 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 sidebar 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 `/lightning` URL. The `/lightning` 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 `/lightning`, 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 `/lightning`, 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 `sidebar` 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,<sup>\*</sup> 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 the Salesforce app. 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](#).

## Chapter 34 Exploring the Visualforce App Container

---

\* There is no Salesforce cantina. Alas.

# CHAPTER 35 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 the Salesforce app.

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 the Salesforce app.

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`—Salesforce mobile app theme
- `Theme4u`—Lightning Console 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

## Chapter 35 Sharing Visualforce Pages Between Classic and Lightning Experience

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>
    <!-- Salesforce mobile theme -->
    <apex:variable var="uiTheme" value="SalesforceApp"
        rendered="{!$User.UIThemeDisplayed == 'Theme4t'}">
        <apex:stylesheet value="{!URLFOR($Resource.AppStyles,
            'mobile-styling.css')}" />
    </apex:variable>
```

```
</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.

To detect what the user sees in JavaScript, we use a similar method to determining the current user experience context in Visualforce. Call the `UITheme.getUITheme()` global variable to return a value that identifies the current user interface theme.

Here the code checks if the current user experience context is the Lightning Experience theme.

```
function isLightningDesktop() {
  return UITheme.getUITheme === "Theme4d";
}
```



## 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`—Salesforce mobile app theme
- `Theme4u`—Lightning Console 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 36 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 the Salesforce app) 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
<code>back ([refresh])</code>	Navigates to the previous state that’s saved in the <code>sforce.one</code> history. It’s equivalent to clicking a browser’s Back button.
<code>navigateToSObject (recordId[, view])</code>	Navigates to an sObject record, specified by <code>recordId</code> .
<code>navigateToURL (url[, isredirect])</code>	Navigates to the specified URL.
<code>navigateToFeed (subjectId, type)</code>	Navigates to the feed of the specified <code>type</code> , scoped to the <code>subjectId</code> .
<code>navigateToFeedItemDetail (feedItemId)</code>	Navigates to the specific feed item, <code>feedItemId</code> , and any associated comments.

Function	Description
<code>navigateToRelatedList ( relatedListId, parentRecordId)</code>	Navigates to a related list for the <code>parentRecordId</code> .
<code>navigateToList ( listViewId, listViewName, scope)</code>	Navigates to the list view that's specified by the <code>listViewId</code> , which is the ID of the list view to be displayed.
<code>createRecord ( entityName [, recordTypeId])</code>	Opens the page to create a new record for the specified <code>entityName</code> , for example, "Account" or "MyObject__c".
<code>editRecord (recordId)</code>	Opens the page to edit the record specified by <code>recordId</code> .

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 the Salesforce app) 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 `navigateToObject(recordId)`

There are actions and functions for viewing, creating, editing, and so on. Use them, rather than URL strings.



## CHAPTER 37 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.

## Chapter 37 Understanding Important Visual Design Considerations

- 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 38 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 Salesforce Mobile-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 `sidebar` 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 `sidebar` attributes of `<apex:page>` have no effect on Visualforce pages when displayed in Lightning Experience.

Pages behave as though the `showHeader` and `sidebar` 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 Salesforce Mobile-Only

---

The `sforce.one` JavaScript utility object is available to Visualforce pages in both the Salesforce app 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, the Salesforce app, 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 most standard and all custom objects 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, along with some elements of object tab, like recent items. Other elements, 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	Salesforce App
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	record create

<b>Override in Setup</b>	<b>Salesforce Classic</b>	<b>Lightning Experience</b>	<b>Salesforce App</b>
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 Hierarchy [81](#)
- Account Insights [42](#), [100](#)
- Accounts [8](#), [81](#)
- Actions [137](#)
- Activity Timeline [76](#)
- Apex [38](#)
- App Launcher [55](#)
- App Manager
  - overview [152](#)
- App Menu [55](#)
- AppExchange apps [37](#)
- Apps
  - lightning [145–146](#), [152](#)
  - overview [145–146](#), [152](#)
- Assistant [100](#)
- Attachments [111](#)

## C

- Calendar [116](#)
- Case feed [105](#)
- Cases [105](#)
- Chatter [108](#)
- Classic apps
  - upgrade to Lightning apps [161–162](#), [164](#)
- Clean up your Salesforce implementation [243](#)
- Compact layouts [131](#), [134](#)
- Comparing the two interfaces [3](#), [21](#)
- Contact Hierarchy [81](#)
- Contacts [8](#), [81](#)
- Contacts to Multiple Accounts [81](#)
- Creating Lightning Experience apps [155–157](#)
- Custom apps
  - lightning [145–146](#), [152](#), [155–157](#)

- Custom buttons

- JavaScript [203–204](#), [207](#), [209](#), [214–216](#), [219](#), [221](#), [228](#)

- migrate functionality to Lightning Experience [203–204](#), [207](#), [209](#), [214–216](#), [219](#), [221](#), [228](#)

- support in Lightning Experience [203–204](#)

- Custom help links [64](#)

- Custom objects [105](#)

- Custom URL buttons [214–215](#)

- Custom Visualforce buttons [216](#)

- Customization

- programmatic [38](#)

- Customizing [55](#), [134](#)

- Customizing Lightning Experience [131](#)

## D

- Documents [111](#)

- Duplicate management [42](#)

## E

- Enabling by edition [43](#)

- Enabling Lightning Experience [45](#)

- Enabling users [43](#)

- Enhanced Notes [42](#)

- Evaluate your org's readiness for Lightning Experience [38](#)

- Events [116](#)

## F

- Feed [108](#)

- Field Service Lightning [94](#)

## G

- Get your organization ready [42](#)

- Global search [58](#)

## Index

Global variables

  \$User.UITheme [320](#)

  \$User.UIThemeDisplayed [320](#)

## H

Help menu [64](#)

Highlights panel [134](#)

Home [8](#), [100](#)

## J

JavaScript custom buttons

  security issues [204](#)

## L

Lead conversion [42](#)

Lead Workspace [80](#)

Lightning actions

  3rd-Party API Integration [228](#)

  populate fields based on user input [221](#)

  provide feedback messages to users [221](#)

Lightning Apps

  create [155–157](#)

  navigation bar [146](#)

  overview [145–146](#), [152](#)

  upgrade from Classic app [161–162](#), [164](#)

Lightning component actions [219](#), [221](#), [228](#)

Lightning Experience

  JavaScript custom button support [203–204](#)

Lightning Experience Developer Edition [244](#)

Lightning Experience features chart [236](#)

Lightning Experience overview [1](#), [7](#), [17](#)

Lightning Experience Setup [74](#)

Lightning Experience Supported Editions [20](#)

Lightning Experience Supported User Licenses [20](#)

Lightning Knowledge [89](#)

Lightning Service Console [88](#)

Lightning Setup page [45](#)

Limitations [72](#), [103](#)

List views [79](#), [103](#)

List Views [8](#)

## N

Navigation menu [51](#)

## O

Object home [76](#), [80](#), [103](#)

Object Manager [72](#)

Omni-Channel [95](#)

Opportunity home [79](#)

Opportunity Kanban [8](#), [79](#)

Opportunity Workspace [8](#), [76](#)

Optimize Lightning Experience

  Drive user adoption [262](#)

Other enhancements [99](#)

## P

Page layouts [131](#)

Path [80](#)

Performance chart [100](#)

Permission sets [43](#)

Piloting Lightning Experience to your users [43](#)

Preview Lightning Experience [39](#)

## Q

Quick actions [209](#)

## R

Recommended features [42](#)

Reports and Dashboards

  dashboard components [124](#)

  dashboard editor [123](#)

  feature comparison [128](#)

  interactive charts and filters [125](#)

  interactive dashboards [124](#)

  new home pages [122](#)

  overview [121](#)

  report run page [125](#)

Rolling out Lightning Experience

  assess your findings [244](#)

  communication plan [258](#)



Rolling out Lightning Experience (*continued*)

- crafting a communication strategy [258](#)
- crafting a marketing strategy [258](#)
- create a Chatter group [254](#)
- creating a project schedule [256](#)
- creating a training plan [259](#)
- decide who to roll out to [248](#)
- define measures for success [257](#)
- deliver an executive summary [267](#)
- demo [240](#), [246](#)
- educate your company [262](#)
- educating your company [239–241](#), [244](#), [246](#), [252](#)
- Educating your company [245](#)
- educating yourself [236](#), [241](#), [244](#)
- employee survey [264](#)
- enabling Lightning Experience [261](#)
- engage with super users [254](#), [262](#)
- engaging your super users [252](#)
- executive sponsor [239](#), [252](#), [254](#), [267](#)
- gap analysis [241](#), [244](#)
- go live [236](#), [246–247](#), [255](#), [258](#), [260–262](#), [264](#), [267–268](#)
- identifying stakeholders [239](#)
- Lightning Setup page [261](#)
- measure results [257](#)
- measuring results [264](#), [266](#)
- overview [235](#)
- phased rollout [248](#)
- picking a launch date [255](#)
- planning your rollout [248](#)
- ready to start [260](#)
- revisiting your processes [240](#)
- risk severity matrix [244](#)
- rollout checklist [248](#)
- rollout strategy [236](#), [246–248](#), [252](#), [254–256](#), [258–262](#), [264](#), [266–267](#)
- rollout structure [248](#)
- super users [252](#), [262](#)
- test your customizations [260](#)

Rolling out Lightning Experience (*continued*)

- useful Trailhead modules [236](#)
- where to start [248](#)

Rollout overview [248](#)

## S

- Sales in Lightning Experience [75](#)
- Sales path [42](#)
- Sales Path [76](#)
- Salesforce Classic [3](#), [19](#)
- Salesforce Files [111](#)
- Salesforce Optimizer [243](#)
- Search [58](#)
- Service Analytics app [93](#)
- Service in Lightning Experience [87](#)
- Service Setup Flows [74](#)
- Setup [51](#), [55](#), [58](#), [64](#), [72](#)
- Shared Activities [42](#)
- Snap-ins chat [90](#)
- Social Accounts, Contacts, and Leads [42](#)
- Social Customer Service [96](#)
- Supported vs. unsupported features [21](#)
- Switcher [47](#)
- Switching between Lightning Experience and Salesforce Classic [47](#)

## T

- Tasks [116](#)
- Turn it on! [45](#)

## U

- Understanding both user interfaces [19](#)
- Unsupported features [37](#)
- Upgrade Classic Apps
  - overview [161](#)
  - task [162](#)
  - test [164](#)
- User profiles [105](#)

## Index

### V

Visualforce 38, 279–281, 283, 297, 299, 305–306,  
310–311, 313–315, 317, 319–320, 322–324,

327–330, 333–335, 337, 339–342

### W

What's new 75, 87