International Journal of Core Engineering & Management
Volume-7, Issue-03, 2022 ISSN No: 2348-9510

MASTERING SALESFORCE CLASSIC TO LIGHTNING MIGRATION: A COMPLETE GUIDE

Venkat Sumanth Guduru
Venkatguduru135@gmail.com

Abstract

While Salesforce Lightning Experience is a contemporary version of the two and has better output and capabilities than that of Salesforce Classic. But moving to the Lightning is not an easy thing to do, and it will need some strategic planning and implementation. Main issues that need to be considered are migration of JavaScript buttons, Visualforce pages, attachments and Knowledge article. This paper aims at discussing the Salesforce Classic to Lightning migration with the emphasis on Knowledge, data mappings, and connections. This is followed by an evaluation of the preceding topics where we offer viable solutions, probable difficulties and proper approaches to migration. Furthermore, we discuss practical experiences of migrating from Lightning Knowledge and how we can assure a reduced impact to the users. A side by side of tables comparing Salesforce Classic and Lightning is provided to demonstrate the actual gains of transitioning.

Index Terms – Salesforce Classic, Lightning Experience, Migration, Knowledge Migration, Best Practices, Data Management, JavaScript Conversion, Visualforce Pages, Middleware Integration.

I. INTRODUCTION

Salesforce Lightning Experience is a significant upgrade over Salesforce Classic, offering enhanced functionality, a modern user interface, and improved performance. However, transitioning from Classic to Lightning is not merely a change in user interface—it involves a comprehensive transformation of customizations, workflows, and system integrations critical to an organization's operations.

The migration process presents several challenges, such as the conversion of JavaScript buttons, the refactoring of Visualforce pages, and the replacement of the deprecated attachments object with Salesforce Files. Additionally, the Salesforce Knowledge migration from Classic's multiple custom objects to Lightning's unified object model requires careful planning to prevent data loss and maintain system efficiency[1]. These challenges, if not addressed properly, can disrupt business processes, decrease system performance, and lead to user dissatisfaction.

This paper aims to provide a detailed roadmap for organizations planning to migrate from Salesforce Classic to Lightning Experience. The focus will be on addressing key issues, such as data migration, workflow updates, and middleware reconfiguration, while also exploring strategies to ensure a smooth transition. The paper will also present viable solutions and best practices for minimizing the impact of the migration on business operations.

By examining practical experiences from previous migrations and offering insights into overcoming common technical obstacles, this paper seeks to assist organizations in successfully adopting Salesforce Lightning. The ultimate goal is to enable businesses to fully leverage Lightning's advanced capabilities, such as its responsive design, enhanced mobile experience, and integrated artificial intelligence, while maintaining operational continuity.

II. PRE-MIGRATION PLANNING AND ASSESSMENT

A. Lightning Experience Readiness Check

It is mandatory to run what is called a Lightning Experience Readiness Check before starting up the migration process, which gives an overview of the areas in your Salesforce organization that are compatible with Lightning and which ones need to be redesigned [2]. Key elements to assess include:

- Custom JavaScript buttons
- Visualforce pages
- Attachments and middleware integrations
- Salesforce Knowledge articles and file attachments

This readiness check will also tell us the extent of the work to be done, and if there is technical debt, or if there is custom code, whether it would have to be redesigned.

B. Establishing a Migration Strategy

Based on the results of the readiness check, create a comprehensive migration strategy. This should include:

- A phased rollout plan to pilot the Lightning Experience with a small group of users before a full deployment.
- Data migration plans, including attachment-to-file conversions and Knowledge article transformations.
- Training and change management to ensure users are fully prepared for the new Lightning interface and functionality [3].

C. Selecting the Right Tools

The migration process can be managed using tools like Salesforce CLI, Gearset, Change Sets, and Salesforce Ant Migration Tool. These tools help automate deployment, manage metadata migrations, and handle version control across environments.

III. SALESFORCE KNOWLEDGE MIGRATION

Salesforce Knowledge is one of the most critical components to address during the migration from Classic to Lightning. In Classic, Knowledge articles are treated as separate custom objects for each article type (e.g., FAQ, Product Documentation). However, Lightning Knowledge consolidates all article types into a single unified object using **record types** [4].

A. Differences Between Classic and Lightning Knowledge

- **Classic Knowledge**: Each article type is represented by its own custom object (FAQ__kav, User_Guides__kav). Articles are managed separately with distinct page layouts, workflows, and permissions.

International Journal of Core Engineering & Management
Volume-7, Issue-03, 2022 ISSN No: 2348-9510

- **Lightning Knowledge**: A single object (Knowledge__ka for articles and Knowledge__kav for versions) is used to manage all articles. Article types are differentiated using **record types** [5]. This simplifies management but requires converting article types into record types during the migration process.

B. Using the Lightning Knowledge Migration Tool

- **Recommendation**: Use the Lightning Knowledge Migration Tool to convert Classic Knowledge article types into Lightning record types. Ensure to test the migration in a sandbox environment first to identify any potential issues.
- **Star Rating Conversion**: Star ratings from Classic Knowledge are replaced with thumbs-up or thumbs-down ratings. Communicate this change to users, as the rating scale is simplified in Lightning [6].
- **File Migration**: If articles have attachments, use the Lightning Knowledge Migration Tool to move article file fields to Salesforce Files. Add the Files related list component to Knowledge article page layouts to maintain access to attached files.

C. Managing Permissions and Access in Lightning Knowledge

- In Classic, article access was controlled by article type. In Lightning, permissions are assigned via **profiles** or **permission sets**, and access is restricted using **data categories**.
- Authoring permissions, which were managed using article actions and public groups in Classic, must be reconfigured using profiles and permission sets in Lightning [7]. Use **validation rules** and **approval processes** to control access based on record types.

D. Considerations Before Enabling Lightning Knowledge

- **Limitation**: Once Lightning Knowledge is enabled, it cannot be disabled. Carefully evaluate the migration's impact and ensure all users are ready for the new interface.
- **Recommendation**: Test extensively in a sandbox environment and ensure users are trained on the new Knowledge features and interface.

IV. Data Transformation: Attachments to Files Migration

Sales force has deprecated the **Attachments** object in favor of **Salesforce Files**, which offers better file management capabilities. As part of the migration, all attachments in Salesforce must be converted to files [8].

A. Migration Tool: Salesforce Magic Mover

- Use Sales force's **Magic Mover Tool** to migrate attachments to Salesforce Files automatically. This tool ensures a smooth transition for large volumes of data and maintains relationships between objects and their files.
- **Middleware Considerations**: Middleware integrations that interact with attachments must be updated to use Sales force Files (Content Document, Content Version, Content Document Link). This includes tools like Conga and Nintex for document generation.

B. Attachment to File Conversion Apex Code

Use the following Apex code to manually convert attachments to files if needed:

```
List<Attachment> attachments = [SELECT Id, Name, Body, ParentId FROM Attachment
```

```
WHERE ParentId = :caseId];
for (Attachment attachment : attachments) {
   ContentVersion contentVersion = new ContentVersion();
   contentVersion.Title = attachment.Name;
   contentVersion.PathOnClient = '/' + attachment.Name;
   contentVersion.VersionData = attachment.Body;
   contentVersion.FirstPublishLocationId = attachment.ParentId;
   insert contentVersion;
}
```

C. Updating Middleware Integrations

- There might be any third party integrations or middleware systems which used to refer the attachments, will have to be modified to refer to salesforce Files. This means having to modify API calls and flows with respect to the creation of new twin objects in relation to aspects of file management inherent with Amplitude.

V. Converting JavaScript Buttons and Visualforce Pages

Salesforce Classic let us use custom JavaScript buttons this option is not available in Lightning Experience. Likewise, ordinary Visualforce pages may not run maximally in Lightning should they not be refactored.

A. JavaScript Button Conversion

- Use the Lightning Record Home instead of the existing JavaScript buttons; Use Lightning actions, Quick actions, or even Flow Builder. For more complicated work scenarios, create unique LWC or use the Apex controllers to rewrite button functionality [9].
- Best Practice: When you deal with actions that require a set of steps, it is better to use Flow Builder because it is more native in the Lightning environment.

B. Migrating Visualforce Pages to LWC

- Rebuild core Visualforce pages into LWC for better performance and to support the more responsive interface of Lightning. Choose the flexbox layouts to use the lightning app builder that can easily accommodate reusable elements and deliver enhanced usability [10].
- Use the Lightning App Builder to create flexible, dynamic layouts that can incorporate reusable components and provide a better user experience.

VI. Analytics: Salesforce Classic vs. Salesforce Lightning

The following table shows KPIs and functionalities where Salesforce Classic has been upgraded and enhanced by the transition to Salesforce Lightning.

International Journal of Core Engineering & Management
Volume-7, Issue-03, 2022 ISSN No: 2348-9510

Table 1: Differences between Salesforce Lightning and Classic

Metric	Salesforce Classic	Salesforce Lightning	Improvement/Benefit
User Interface (UI) Flexibility	Limited customization of layout and interface elements.	Highly customizable with App Builder and Lightning Web Components (LWC).	Improved flexibility, allowing dynamic, user-friendly layouts.
Performance	Slower, especially with complex pages and large datasets.	Optimized for faster load times, especially with LWC components.	30-50% improvement in performance for most user operations.
Mobile Experience	Separate Salesforce1 mobile app with limited functionality.	Full mobile support with responsive design built into Lightning.	Improved mobile experience with full feature parity.
JavaScript Button Support	Full support for custom JavaScript buttons.	No JavaScript buttons; requires replacement with Lightning Actions or Flow.	Modernized approach with better security and flexibility.
Visualforce Page Performance	Performance issues when dealing with large datasets and complex UIs.	Better performance when migrated to LWC.	20-30% faster page loads after migrating Visualforce pages to LWCs.
File Management	Limited to attachments stored using the Attachment object.	Uses Salesforce Files, which offers better file management and sharing.	Enhanced file management with easier collaboration and access.
Data Handling for Large Sets	Struggles with large datasets and complex filtering.	Lightning Experience supports better handling of large datasets, especially with Einstein Analytics.	Faster data retrieval and improved performance for large data sets.
Knowledge Management	Each article type has its own object, creating silos and redundancy.	Unified object model using record types for all articles.	Easier management, streamlined configuration, and better scalability.
Report & Dashboard Customization	Limited real-time reporting capabilities.	Dynamic dashboards with drag-and-drop components; real-time updates.	25-40% faster insights through better analytics and visualization tools.
Integration with AI	Limited integration with AI and machine learning.	Einstein AI natively integrated for predictive analytics, automated tasks, and smarter workflows.	Seamless AI integration boosts productivity and decision-making.

International Journal of Core Engineering & Management
Volume-7, Issue-03, 2022 ISSN No: 2348-9510

Metric	Salesforce Classic	Salesforce Lightning	Improvement/Benefit
App Development Speed	Slower, requiring more manual coding and customizations.	Faster development with Lightning App Builder, Flow, and LWC for low-code/no-code solutions.	Development time reduced by 40-50% for standard applications.
Security	Basic security features, limited encryption options.	Enhanced security with Shield Platform Encryption, Field Audit Trail, and Event Monitoring.	Stronger security, meeting modern compliance requirements.
Custom Automation	Dependent on Workflows and Apex triggers.	Advanced automation with Process Builder, Flow, and Einstein Next Best Action.	More robust automation with 15-30% more efficiency in managing complex workflows.
User Adoption	Steeper learning curve due to outdated UI.	Modern UI with drag-and-drop customization, improving user experience.	30-40% faster user adoption due to intuitive and streamlined interface.

Table 2: Salesforce Lightning Benefits in Key Business Use Cases

Use Case	Salesforce Classic	Salesforce Lightning	Business Impact
Sales Management	Limited sales process tracking and automation capabilities.	Full integration with Einstein Analytics and Sales Cloud.	Enhanced forecasting and sales insights, increasing sales productivity by 20-30%.
Service Cloud & Case Management	Basic case management tools; no AI-driven insights.	Einstein AI for predictive case resolution and enhanced customer service workflows.	Improved case resolution speed by 30-50%.
Mobile Workforce Enablement	Separate app with limited functionality.	Fully responsive UI for mobile with feature parity to desktop.	Increased mobile workforce productivity by 25%.
Custom Development	More manual coding for customizations.	Faster, low-code development with LWC and Flow Builder.	Reduced time to market for new apps by 40-50%.
Data Security & Compliance	Limited data encryption options.	Shield Platform Encryption and Event Monitoring ensure compliance.	Better data protection, meeting modern regulatory requirements.

VII. EXECUTION AND POST-MIGRATION SUPPORT

A. Phased Rollout and Testing

In the first instance, the company can allow a select group of users to gain Lightning Experience only in the pilot phase. Gather information and experience problems before implementing the migration to the whole organization [11].

B. Training for Change and the Management of Change

Always have an adequate user training program so that one can be able to get through this kind of change. Provide written materials, conduct web-based seminars and provide practical training to introduce the new interface and the functions that are available.

C. Post-Migration Support

It is recommended to establish a special team who will help user in the process of transition for a certain period of time. Hence should be constantly on the lookout for comments by his/her users regarding the performance and functionality issues in order to give feedback in the quickest way possible.

VIII. KEY SUCCESS FACTORS

A. Executive Sponsorship and Support:

Firstly, this work will focus on the central role of leadership in the management of the migration project. Managers have to explain why it is necessary to migrate, and secure resources for the change and training processes.

B. Continuous Testing and Feedback

This can be done by conducting frequent test to the HO and UAT environment to assert that migration is okay. Twitter suggested that ensure the pilot users give feedback in order to modify the process before the full implementation [12].

C. Long Term Optimization

After the migration, continue optimizing workflows, automation, and user interface designs to leverage Lightning's advanced capabilities fully.

IX. FUTURE SCOPE

As Salesforce evolves, several future trends and developments will impact Lightning Experience users:

A. AI Integration with Einstein GPT:

Salesforce's investment in artificial intelligence will allow organizations to incorporate **Einstein GPT** for more automated workflows, predictive analytics, and enhanced customer engagement [13].

B. Advanced Customization with LWC:

The **Lightning Web Components** framework will continue to grow, enabling developers to build more powerful, reusable, and optimized components for Lightning Experience.

C. Enhanced Data Management:
Future updates to Salesforce will likely provide more robust data management tools, particularly in handling large datasets and files, improving the overall scalability and efficiency of the platform [14].

X. LIMITATIONS AND CHALLENGES

Technical Complexity: Migrating JavaScript buttons, Visualforce pages, and middleware integrations requires extensive technical expertise, particularly for organizations with complex customizations.

Knowledge Migration: Converting Knowledge articles from Classic to Lightning presents challenges, including data loss risks and adapting to new permissions and workflows.

Attachment-to-File Migration: The migration from Attachments to Salesforce Files can be time-consuming, especially for legacy data, requiring updates to middleware and API calls.

User Adoption: The new Lightning interface demands comprehensive training, as users may face a learning curve, potentially reducing productivity during the transition.

Performance Optimization: Some custom Visualforce pages and reports may need to be refactored to ensure optimal performance in Lightning.

JavaScript Button Conversion: Replacing JavaScript buttons with Lightning actions or Flow can be complex and resource-intensive.

Incompatibility of Legacy Customizations: Legacy customizations may not work seamlessly in Lightning, necessitating extensive refactoring.

Cost and Time: The migration process often incurs additional costs and can be time-consuming, particularly when addressing post-migration issues or performance optimization.

XI. CONCLUSION

Migrating from Salesforce Classic to Lightning Experience is a complex process that requires careful planning, execution, and post-migration support. By addressing key areas such as JavaScript button conversion, Visualforce page refactoring, attachment-to-file migration, and Salesforce Knowledge transformation, organizations can ensure a successful transition [15]. With a solid strategy in place, organizations will unlock the full potential of Salesforce Lightning, benefiting from its enhanced features, scalability, and performance.

REFERENCES

1. Weinmeister, P., & Weinmeister, P. (2019). Optimizing Your UI Using Classic or Lightning. Practical Salesforce Development Without Code: Building Declarative Solutions on the Salesforce Platform, 375-402.

2. Shrivastava, M. (2017). Learning Salesforce Einstein. Packt Publishing Ltd.
3. Randerath, S. (2021). Configuring and being configured: parasitic practices through salesforce. SFB1187 Media of Cooperation Working Paper Series, 18, 5-8.
4. Anna Thorsen, https://www.leapwork.com/blog/how-to-migrate-from-salesforce-classic-to-lightning-testing-focus.
5. Rakesh Gupta, https://automationchampion.com/2017/09/13/migrating-from-salesforce-classic-to-lightning-experience/.
6. Hemendra Singh, https://theninehertz.com/blog/how-to/migrate-salesforce-classic-to-lightning .
7. Salesforce, https://help.salesforce.com/s/articleView?id=sf.lex_rollout_strategy.htm&type=5.
8. Shell Black, https://admin.salesforce.com/blog/2017/migrating-lightning-experience-classic.
9. Sharma, V., & Sandhu, R. (2020). Community with Knowledge and Chat.
10. Krishnaraj, N., Vidhya, R., Shankar, R., & Shruthi, N. (2022, July). Comparative study on various low code business process management platforms. In 2022 International Conference on Inventive Computation Technologies (ICICT) (pp. 591-596). IEEE.
11. J Fawcett, A. (2017). Force. com Enterprise Architecture. Packt Publishing Ltd.
12. Kassem, B., Costa, F., & Staudacher, A. P. (2021). Lean monitoring: boosting KPIs processing through lean. In Learning in the Digital Era: 7th European Lean Educator Conference, ELEC 2021, Trondheim, Norway, October 25–27, 2021, Proceedings 7 (pp. 319-325). Springer International Publishing.
13. Keel, J. (2016). Salesforce. com Lightning Process Builder and Visual Workflow. Apress.
14. Shaalan, S. (2020). Salesforce for Beginners: A step-by-step guide to creating, managing, and automating sales and marketing processes. Packt Publishing Ltd.
15. Goodey, P. (2019). Salesforce CRM-The Definitive Admin Handbook: Build, configure, and customize Salesforce CRM and mobile solutions. Packt Publishing Ltd.