

Lawson Application Upgrade from 8.1.3 to 9.0.1 Case Study

Project: Application Upgrade from 8.1.3 to 9.0.1

Industry: Healthcare

Business Challenges:

This healthcare client decided to upgrade their Lawson applications from 8.1.3 to 9.0.1. The driving factor behind the upgrade was to migrate from the Time Accrual (TA) module to Absence Management. For this to be a successful upgrade, it required all of their cloned and modified Lawson programs to be migrated to the next application version. This also required changes to database table definitions, shell scripts that would execute when certain programs were run, the addition of reports and variables for each program, and additional program logic that reflected business requirements that needed to be accommodated.

In addition to the core programs that were to be migrated, the client had also modified each Self Service module within Lawson Portal: Employee, Manager, and Requisition Self Service. By obtaining the newest versions of the Self Service applications, all customizations made in earlier versions would have to be re-implemented.

While the initial scope of the project was strictly related to the Lawson application upgrade, the client expanded the scope. At the request of auditors, they were required to make changes to their security processes. This included restricting all users, other than Change Management employees, from compiling programs in the production environment, reporting all security access and given to managers on a quarterly basis for review, and limiting access to the 'lawson' and UNIX logins to the Lawson system.

High Level Business Solution:

Migrating and updating customized Lawson programs required that Absence Management training be attended by the Subject Matter Experts (SMEs) so that they could determine how to replace Time Accrual applications. All custom interfaces to third party applications would need to be redeveloped to form to the requirements of Absence Management.

With any release of applications, there are always changes and updates to processing logic. These changes needed to be documented and determined if they were required for this health organization was to maintain their business model. In addition, it was also necessary to determine if the changes would eliminate the need for existing customizations that were in place with older versions of the software. All custom programs were reviewed with users from their respective departments and the SMEs to agree on programs to migrate and programs that could be retired.

The security requirements required by the auditors shaped new challenges. A majority of the technical resources at the client had multiple responsibilities, regardless of which specific system they were working on. For example, a system administrator for Lawson would be in charge of applying patches to the system and applications, creating and adjusting security for users, updating the Oracle database if changes were to be made, and responding to production support requests that would require compiling in the production environment. To the auditors, each of these tasks should fall under a different department, rather than an individual.

The approach we took was to delegate these tasks to their respective departments. While this would create additional support requests in their system and possibly delay work that would need to be done immediately, it maintained the auditing standard and also left points of accountability to specific departments. In addition to adopting this model, the client required that each manager review user security of their respective department staff to determine if users required and security adjustments.

Business/Technical Solution Details:

- Migrated custom written programs (interfaces and custom reports) in Lawson. Lawson provides commands (pgmdump and pgmload) that will bundle all pieces of a program into a file so that it can be migrated from one environment to another. After the program definitions are dumped into a file, they are migrated, through FTP, to the new environment, loaded, and then compiled.
- Programs that were cloned from existing programs would need to have any updates to the original code put in place before it was compiled. Cloned programs were moved to the new environment with pgmdump and pgmload. After loaded into the new environment, a program called Beyond Compare was used to analyze the differences between the new base code and the previously modified version. Any changes to base code that were identified were moved to the modified program. After all changes had been identified, the program was compiled and ready for testing.
- Compared any modified base Lawson program to the new release. While maintaining comments and a summary section and using Beyond Compare, identified what custom changes were put in place. After the customizations were placed in the base code, the program was compiled and tested.
- Upgraded the Self Service modules from 9.0 to 9.0.1.4. Identified and migrated customizations.
- Modified Visual Basic code for a GL Journal entry spreadsheet to write the output Excel file into a shared drive rather than requiring users to FTP the files to a server. For security considerations, FTP access was restricted to functional users so that the integrity of the systems wasn't in question.
- Due to the results of the security audit report, created a UNIX shell script that used the ldapsearch command. This command retrieved user roles and identity information from the LDAP server. This information was parsed and reported in a MicroFocus COBOL developed program that would output a different CSV file for each module in Lawson. These reports would be e-mailed to respective managers using a UNIX shell script.
- The security audit requirements also led to the development of an automated report of all the security roles and classes that are used for LSF9 and LAUA security. This program was developed in C#.NET. The initial



SURETY SYSTEMS
Accuracy · Speed · Results

program logic would create an FTP session to retrieve all the XML definitions for each program defined in Lawson. By doing this, all of the function codes for each program could be retrieved and translated. This was a big concern for management because if the function codes were not translated, managers may not know what users have access to. Next, the program queried the LDAP directory to retrieve all the security roles, classes, programs associated with the class, and the function codes that program has access to. This is where the translations were used. After retrieving all the LDAP information, an Excel file was created that would also be sent to managers so that they could cross reference a users' role to what that role translated to. The last section of this program queried the Oracle database to retrieve LAUA security. As a quirk in Lawson, Oracle only stores what users do not have access to. Therefore, the translated values that were retrieved in the initial program logic were used again to find all the valid values a user had access to.

- Another requirement of the auditors was to monitor all compiles that occurred in production. This was also done in C#.NET. The contents of the object directory (compile directory in Lawson) were retrieved and any objects that had been modified in the previous 24 hours were outputted to an Access database setup by Change Management.