

Migrating Lotus Notes Applications to Google Apps



Introduction.	3
Assessment	3
Usage	3
Complexity	4
Action Plan	5
Migration.	5
Integrating applications functionality into Google Apps.	5
Templates and Logic	6
Data	7
Co-Existence	7
Summary and Conclusions	7

Visit <http://www-10.lotus.com/ldd/sandbox.nsf> and search for "UserActivity" to get an example class that allows access to the user activity of a database from LotusScript. The NotesUserActivity class represents a summary of user activity for a specified database as well as providing access to individual user activity entries. User activity for the specified database must be enabled for this function to work. After creating a new instance of the NotesUserActivity object for a database you should integrate the HasUserActivity property to ensure that activity is available.

Introduction

As companies look to move from the legacy Lotus Notes platform to a cloud-based solution such as Google Apps, they have to consider a range of migration strategies for user information, mail, calendar, contact data, applications, and associated data. It's important to classify this migration correctly, as although mail, calendar, and contact data is fairly uniform and can be addressed with general tools such as Google Apps Migration for Lotus Notes, Notes applications tend to vary widely on multiple dimensions.

This document will explore strategies for looking at your Notes applications and their associated data, and determine if and how to migrate this functionality to the Google Apps platform. It will also provide guidelines to support decision-making around migrating specific applications and associated data, based on usage patterns. It will then review the various components of Google Apps that will serve as targets for migrating applications into the Apps platform and discussing strategies for performing those migrations.

Assessment

The first step in any application and data migration is to assess the current situation and evaluate which applications should be moved to the new platform. Often, data repositories and applications get built for a special purpose, and once that event, project, or use is finished, the application lies dormant.

Similarly, applications can grow over time "organically" and become highly complex, such as a simple timecard tracking sheet that becomes a full-blown time and expense reporting application with full approvals and workflows.

Migration to a new platform represents an opportunity for the enterprise to both eliminate or archive unused applications as well as refactor overly complex applications into a more streamlined set of components that better serve their evolved purpose. It is important that both of these dimensions are investigated in the application and data migration.

Usage

Many collaborative applications and databases are used for a point in time event, function, or project, and lie dormant once that event is completed. Little incentive exists for end users or administrators to clean up these applications, and therefore they show as active in the system. What's more, these applications and the data they contain can represent a compliance risk if left unattended. A platform migration is an ideal time to assess the true number of active Notes applications in the enterprise.

In order to assess usage of Notes applications, it is important that you look at both the frequency of update of an application's data as well as the breadth (number) of users that access the application. This paper will not go into the full technical detail of accessing server log or usage history, but this can be done directly against the Domino server or through LotusScript. A variety of third-party tools and consultancies can also assist in performing this type of usage audit.

Once you have reliable usage information, you can classify your applications into the following usage categories based on amount of activity and breadth of activity:

- Inactive: applications that have not been accessed by users in the last 180+ days
- Limited-Active: applications that are used regularly but by a small set of your employee base (less than 10 users, for example)
- Active: applications that are regularly accessed by a number of users

You can use this assessment data to help determine which applications to migrate forward. Applications that are Inactive should be archived in some manner for record, or destroyed and not moved to Google Apps. Applications that are Limited-Active are good candidates for alternative uses or manual user-based migration to Google Apps. Applications that are Active will need a migration plan based on complexity and type of use.

Complexity

The other critical dimension to assess when looking at your Notes applications is the complexity of the application. Complexity of an application can be evaluated on a number of different dimensions. Some typical questions you will ask are:

- How large is the data set?
- How many custom fields are used?
- Is there a great deal of scripted logic to replicate?
- How many forms are used for data input?
- How many views are used for data display?
- How much custom workflow logic and data validation is contained in the application?

Complexity takes into account application design elements, such as scripts, formulas, forms, and agents, that add complexity to an application. The index is calculated on a scale of 1 (lowest) to 5 (highest).

1. Standard applications: Applications that are derived from standard Lotus Notes templates and have not been customized
2. Data applications: Applications that have custom data schemas, but no major customization of views, forms, or logic
3. Configured applications: Applications which consist primarily of custom data schema, views, and forms but no significant business or application logic
4. Scripted applications: Applications which, in addition to custom data schema, views, and forms also include light scripting and logic to perform action on that data
5. Custom applications: Applications that are fully customized including server side logic, application workflow, custom actions, user interfaces and the like

Review each application deployed in the Notes environment and assign it a complexity based on the above scale. When unsure of the appropriate complexity rating, it is advised to rate the next complexity level up. However, it is important to note those more conservative estimates to help determine a better migration estimation with margin of error taken into account.

Action Plan

Based on the above classification of your Lotus Notes applications, you can plan how to move your applications to the Google Apps platform. Determine which applications will be moved to the new platform, what the target for each of those applications will be, and what the turndown plan is for those applications that you don't intend to move. We will also discuss later in the paper the possibility of maintaining a Notes co-existence environment should the return on investment of migrating some or all applications not be viable.

A key component of the action plan is preparing a communication and training plan for your users. Once you have identified targets and executed migration, it is critically important that you train users on how to use the new applications, noting key differences in functionality or user experience.

The final part of the action plan should focus on consolidation and reuse. With a complete, pan-Enterprise inventory of your Notes applications, it is an ideal time to look for commonalities between applications and patterns in usage. Often, applications that are derivatives of the standard Lotus Notes templates and stock applications can be consolidated into a single application type and reused. This not only eases the burden of application migration, but also makes ongoing maintenance and future enhancements more manageable.

Migration

With your inventory and action plan of which Notes applications will be migrated and which will be turned down, the next step is to identify the targets within Google Apps. This consists of mapping the various key functions from within your Notes applications to similar functionality in Google Apps. With targets identified, the next step is performing the application framework and logic migration and any custom development that is required. Once the application targets are fully prepared and tested, you will perform data migration. Some of these processes can be automated based on the number of instances of an application that exist and the relative value of that automation.

Integrating applications functionality into Google Apps

The applications and components built in to the Google Apps suite can both independently and as a system act as the migration "targets" for your Lotus Notes applications. Key components within Google Apps that are relevant to Notes applications are described below.

Google Sites is a site building toolkit for business users. With Google Sites, users can easily create and update their own site. Google Sites allows users to display a variety of information in one place—including videos, seshows, calendars, presentations, attachments, and text—and share it for viewing or editing with a small workgroup, an entire organization, or their whole company. Google Sites offers a set of predefined page types and layouts that act as containers for content and embedded application components in the form of Google Gadgets (see below). It also provides key application functionality such as customizable lists and document "file cabinet" storage, sharing, and versioning. Sites allows full access control capabilities to restrict access to specific users or groups of users, giving you the ability to limit access to applications. Embedded components within Sites can use alternate access control lists, allowing

The Google Secure Data connector provides secure data access between cloud-based applications in Google Apps and a corporate intranet or other internal applications. More information about the SDC can be found here: <http://code.google.com/securedataconnector/>

different users to experience different functionality within a Site (e.g., read only for some users, data entry for others). The power of Google Sites lies in its flexibility and configurability, and most notably in the ability for users to simply embed application components as Google Gadgets.

Google Spreadsheets is a key application for structured data, data manipulation, forms input, and workflow logic. Google Spreadsheets includes the ability to create custom forms to embed within Google Sites or any HTML-based web page or application. It also can perform data input validation as well as sophisticated data analysis and manipulation. For the purposes of application development, one should think about Google Spreadsheets as a powerful database and data management engine with high extensibility.

Google Apps Scripts Google Apps Scripts are a server-side JavaScript interface that give you access to data and functions from across the Google Apps suite. This scripting interface provides rich extensibility for building custom business logic and workflow rules to interact with data brought into Google Apps through Google Spreadsheets. Scripts have access to a rich set of interfaces across Google Apps products including calendar, email, documents, spreadsheets, etc. Scripts can also access external data and web services both from the public Internet (e.g. Google.com) as well as from within your intranet through use of the Secure Data Connector.

Google Gadgets are simple HTML and JavaScript applications that can be embedded in web pages and other apps. A number of pre-built gadgets, provided by Google, expose various application functionality such as a data input form, a flash presentation, a video, and a number of others. In addition, tens of thousands of gadgets built on the Google Gadgets platform are made available through Google Sites. Most importantly, Google Gadgets acts as the development extensibility mechanism for building custom application functionality and business logic that can be inserted into a Google Site or Google Spreadsheet. Gadgets can provide custom user interface elements, advanced business logic, and data manipulation capabilities. Once you develop a gadget, it can be published into a private directory accessible within your enterprise's Google Apps installation, and reused across a number of applications. More information about Google Gadget development can be found at <http://code.google.com/apis/gadgets/>

Google App Engine is a robust development platform providing a fully hosted development environment for creating rich and sophisticated web applications. From Google App Engine, an IT developer can build complete custom applications and application components (including building and hosting Google Gadgets on App Engine) to meet the needs of the most demanding and complex custom Notes application. More details on Google App Engine can be found at <http://code.google.com/appengine/>

When identifying the target for a particular Notes application, it's important to assess whether the application will move to a single product such as Google Sites or Google Spreadsheets, or a combination of products such as Google Sites with embedded Google Spreadsheets forms leveraging Google Apps Scripts for workflow and application logic.

Templates and Logic

Upon mapping the functionality in your Notes applications to Google Apps, the next step is to migrate the application frameworks and associated logic to Google Apps.

This set of activities consists of anything from building out templates and page types within Google Sites, creating data schemas in Google Spreadsheets, building logic and workflow scripts in Google Apps Scripts, and building custom application components in Google Gadgets and Google App Engine.

Data

Once the framework and logic for the applications has been created in Google Apps, you will then move the data. A number of procedures and tools simplify data extraction from Lotus Notes and import into Google Apps. However, it is important to note that the migration is an ideal time to ensure that historical data is archived if it is not being used, and data is cleansed and normalized where appropriate.

Based on where the targets are for various applications, different tools can be used for importing data into Google Apps. Import APIs for Google Spreadsheets and Google Sites are the two primary mechanisms for moving data into Google Apps.

Co-Existence

Some organizations, particularly those with complex installations or usage models, choose to continue to support a limited number of Lotus Notes applications after the move to Google Apps. If the migration costs don't justify an appropriate return on investment, or users and business processes depend on a small set of Notes applications in a way that complicates migration, then continuing to support a Lotus Notes installation may be the logical path.

Once you have migrated your users' email, calendar, address book, and a majority of applications off your Lotus platform, you will likely find that you can reduce licensing costs and turn down a number of the servers in your Notes infrastructure.

Summary and Conclusions

Google Apps is a powerful, extensible, cloud-based application platform that enables a software-as-a-service platform for email, calendar, contacts, office productivity and employee applications, and also supports the implementation of custom line-of-business applications. Over the years, companies have invested significantly in Lotus Notes applications based on the Domino platform. These applications can be adapted to Google Apps as needed with a Lotus Notes migration, often saving ongoing IT maintenance and operational costs.

This paper has reviewed the various aspects of migrating Lotus Notes applications into the Google Apps cloud, and presented a model for assessing and executing those migrations. It also talks about key items to consider such as usage and consolidation in making migration determinations. Finally, it has provided insight into the various Google Apps components that serve as the migration targets for Notes applications and how to think about the movement as well as links to more information about those components.

