Application Program Interface (API)

A path to seamless software integration

By: *Sumanth Bail*

Overview

Electronic Document Management Systems (EDMS) are being used by companies to organize and streamline their document processes to increase productivity and efficiency. Many of these companies also use customized software that is specific to their industry and business needs. In many cases this software is written and maintained by in-house programmers or consultants. Integration between the EDMS and the customized software is much desired because of the improved efficiencies and savings this effort can produce. Integration on a very basic level will prevent duplicate data entry and on a much deeper level might involve document creation or workflow. The integration can be accomplished using an Application Program Interface (API) that comes along with the EDMS Application. (http://www.cabinetng.com/white-papers/software_integration.php)

What is an API?

Definition 1:

An application programming interface (API) is a set of functions, procedures, methods, classes or protocols that an operating system, library or service provides to support requests made by computer programs. (http://en.wikipedia.org/wiki/API)

Definition 2:

An application program interface (API) is the specific method prescribed by a computer operating system or by an application program by which a programmer writing an application program can make requests of the operating system or another application. (http://searchexchange.techtarget.com/sDefinition/0,,sid43_gci213778,00.html)

General:

In a layperson's terminology, an API makes it easier to develop a program, by providing all the necessary building blocks. A programmer then puts the blocks together to connect, create and\or gather information from the application to which the API belongs.

Most EDMS solutions, such as CNG-SAFE, offer an API so that programmers can write add-on applications that are consistent with the core application. Although APIs are designed for programmers, they ultimately benefit users because they help ensure that all programs using the same API set will have similar interfaces. This makes it easier for users to learn and use these modules.

When can an API be used?

Many customers use their EDMS right off the shelf. But some have the need to integrate the EDMS to their existing software or have other specific needs that may require some degree of customization. The considerations for integration or customization include, but are not limited to, process time, cost, savings in man hours, double entry, task scheduling, complexity, consistency and need for the elimination of human error, and sensitivity of a given task.

When does it make sense to automate? Generally, when we can save time and man hours. There are a few scenarios provided below (see sample usages) which show how these savings might be accomplished. But it is not always about time and hours saved. To elaborate on this, let's look at a very simple example. Say we have a finance cabinet. The requirement here is for a user to file a few monthly financial statements, generated externally, at the end of each month. It does not take much time and hence does not seem cost effective to automate this process. But what if these documents are regulated and require a retention policy set, so that they are not to be deleted for 7 years, and have to be filed by a certain time period. The consequences of not following these guidelines might turn out to be expensive. This makes the task mission critical and you would not want potential human error of misfiling or not filing. This then provides a case for automation outside the quest of saving time and man hours.

Where can an API be applied?

APIs provide most of the functionalities available in the core application and are controlled to give and deny access depending on user permissions. Using an EDMS API, it is possible to synchronize data ["Synchronizer" (http://cabinetng.com/products/synchronizer.php)], retrieve specific document(s) ["Retriever" (http://cabinetng.com/products/retriever.php)], automate import of documents, route documents, and get user permissions, etc.

General requirements to use an API

- A programmer with at least a little knowledge in the use of APIs.
- An understanding of the current integration project objectives.
- Ability to build with and around the API calls to successfully complete the integration.

Sample usages

The possibilities of adding value to an existing application by integrating with a document management system are endless. The goal is to improve operational efficiencies by complementing the existing application with common document management features such as scanning, filing, and organizing documents in a structured environment. Advanced integrations can perform sophisticated tasks such as automated workflows based on business policies and other criteria shared between applications. Below are a few common examples of applying an API to facilitate custom integrations.

Scenario 1:

The givens: An AS400 system that creates daily text invoice reports based on account numbers. An EDMS exists with a cabinet containing folders for all accounts.

Process without automation: The whole report is filed to a separate reports cabinet. Drawbacks to this process are: a user has to manually file to the appropriate folder and name the document. The document has invoices from multiple accounts. Associated documents for the invoices are in a different cabinet.

The need: The full report has to be split into individual account reports and filed to the appropriate EDMS location

Solution: Write a utility that monitors the report output directory. Split the report based on each invoice. Obtain the account number, then using the API; create a folder if it does not already exist. Obtain the invoice number to name the individual report in the EDMS and file it using the API.

Scenario 2:

The givens: A SQL based medical Practice Management System (PMS). An EDMS with a cabinet containing folders for all patient charts with pertinent information.

Process without automation: User manually creates or updates patient folder with information from PMS before creating or filing any documents into them. The drawback to this is the manual process, the double entry and the potential for incorrect patient information being entered.

The need: Automatically update patient data from the PMS to the patient folders in the EDMS in order to accelerate the process and avoid duplicate entry.

Solution: Write a utility that periodically connects to the PMS database and gets the latest modified patient data. Use the API calls to then create or update the patient folder data in the EDMS.

Scenario 3:

The givens: A client has custom software that can browse through their customer list. They also have an EDMS that has a cabinet, with folders for each customer.

Process without automation: User has both the custom and the EDMS software open. If the user has browsed to a specific customer and needs to view related documents in EDMS, then the search mechanism in the EDMS is used get to the required customer folder.

The need: They would like the ability to be on a customer page in their software and open that specific customer folder in the EDMS.

Solution: Write a function, using the appropriate API calls, to display the EDMS folder for any specified customer.

Conclusion

Using an API makes the external software integration or automation processes with the EDMS feasible and in most case seamless, this by enabling end users to access documents from the comfort of their current application. API integration with the EDMS allows users to perform routine tasks more quickly, with fewer errors, and at a lower cost by making all the documents required to complete these tasks immediately available.