

SOLUTION STORY

BACKGROUND

This customer is a health organization that promotes quality healthcare.

CHALLENGE

The customer needed a solution to manage incoming medical appeals from third parties. They are contracted to handle these appeals and had several primary requirements for PowerObjects. Requirements included a CRM installation and infrastructure setup; business process solidification; capturing incoming appeals in a secure, HIPPA-compliant manner into CRM from documents supplied by third-party organizations to the customer; automating mail merge functionality in order to generate response letters for appeals in order to make workloads more manageable; ability to edit letters before they are sent out; automating appeal (case) routing functionality so that users could click a button and be assigned the oldest appeal in the system that they were qualified to work on; security model to ensure HIPPA compliance; field level security on case fields; unique case number generation; automation for QA routing; and automation for "Send Back" appeals from third parties into separate buckets.

SOLUTION

PowerObjects built out solutions for their needs in CRM and used custom pieces to satisfy the requirements. PowerObjects installed CRM and assisted the customer with setting up the infrastructure; helped the customer outline their business process and documented them via flow charts and word documents; and then utilized a Secure File Transfer Protocol 000(SFTP) between the customer and their third party appeals generator in which the third party would upload documents and the .NET code that PowerObjects created would fire every night to check for documents in the SFTP.

If the code found SFTP, it would run the document through a series of logical checks. If it triggered on one of the logical checks, the code would take the appropriate action on the document, read the document, parse out the information into an appeal entity record in CRM, and then create the appeal entity record. Next, the code PowerObjects wrote would move the document out of the SFTP folder into a "processed" folder.

Additionally, PowerObjects built a case stage and case structure entity for each appeal. Each appeal could be in a single case stage and a single case status at any given time. JavaScript checks the case status and stage onload of the form and also checks the logged in users' security role and team. If a user had the appropriate team/security role combination, as per the case stage/status, they would be able to edit and see fields. If the user did not have the appropriate team/security role combination, they would either not be able to see the record or they would not be able to edit the fields and would only be able to view limited fields. This was laid out in their business process by PowerObjects.

HEALTH ORGANIZATION

LEVERAGES DYNAMICS

CRM TO MANAGE

MEDICAL APPEALS

AWARDS + RECOGNITION

2012 + 2013 + 2015
Microsoft Dynamics Partner
of the Year

+++

2011 + 2013 + 2014
Inner Circle for Microsoft
Dynamics

+++

2009 + 2010 + 2011 + 2012
Inner Circle for Microsoft
Dynamics

+++

3
Convergence Customer
Excellence Awards

+++

GOLD + SILVER
Inner Circle for Microsoft
Dynamics

**POWER
OBJECTS**

Headquartered in Minneapolis

612.339.3355 | 866.770.3355
powerobjects.com

BENEFITS

PowerObjects built buttons that would route appeals to users. PowerObjects created separate buttons for each “Group” that the customer had. For example, if you were in Group ABC, you could click the “Assign me ABC Appeals” button. Once clicked, code would fire and find the oldest possible appeal and assign it to the user.

PowerObjects also built custom buttons for the user to click when they were ready to move the appeal to the next stage. Once clicked, the appeal would be sent to the next feasible stage, based on what information was entered in the form. Button display/enable rules were utilized heavily to drive what each user could view and edit in terms of buttons since different privileges existed for different roles.

PowerObjects built a random-number generator for unique ID numbers and built code that would push 20% of appeals that were marked as “Complete” to a QA bucket, where the appeal would then get reviewed by the QA Team. The QA Team had separate set up buttons and viewing rights that PowerObjects had to construct. Once an appeal was finalized, a plug-in was built that would transfer the information into a PDF letter format and store it in SharePoint. From there, code would fire each night to look for Documents in the SharePoint folder. The code would take the found PDF documents, run them through logical checks, and uploaded them to an outgoing SFTP folder.

This process sent files back to the third party after they had been run through the business and appeal review processes by the customer’s users. If the third party approved the results, that appeal was completed. However, if the third party determined that changes needed to be made, they would send the file back with an “R” appended to it. Code checked for documents marked “R” in the SFTP, find the existing Appeal that had been previously completed in CRM, and re-open it with the “R” appended and with the “R” reason uploaded. The main integration methods for this project were .NET applications and plug-ins.



ABOUT POWEROBJECTS

Established in 1993, PowerObjects is one of a handful of organizations recognized as a leader in delivering Microsoft Dynamics CRM/xRM solutions to customers. PowerObjects fully embraces the power of customer choice and offers hosted, on-premise and online CRM deployment options.

PowerObjects provides CRM solutions to businesses and non-profit organizations in multiple industries including healthcare, life sciences, insurance, financial services, publishing, distribution, manufacturing and professional services.



Headquartered in Minneapolis

612.339.3355 | 866.770.3355
powerobjects.com