

Success Story



THE COMPANY

Collection Partner (formerly known as Hubbard Systems) has been providing high-performance, mission-critical debt collections software applications since 1985. Their application suite handles both the accounting functions of the collections industry and the legal aspects of collecting debt. They provide the highest quality, most comprehensive, and flexible debt collections software available, as well as superior service and support, by listening to their clients, and turning suggestions into system enhancements.

THE NEED FOR A SOLUTION

Collection Partner had been working with their COBOL vendor since their start in the 1980s, but they were ready for a change. Matthew Burke, CTO of Collection Partner, said “It got to the point where we felt we weren’t getting the level of support we needed. And we felt like we were always dealing with issues with their products”. But their main reason for modernization was the reality that their legacy application was limiting from a user-experience and functionality standpoint. Their customers often had employee retention issues, especially among younger employees, due to the user-experience of their application. They also realized that if they were to stay relevant in their industry, they had to offer a better solution, which required a modernized application.

Collection Partner’s first attempt to update their legacy product was in the early 2000s when they attempted to replace it with a SQL-based rewrite. They brought in a big team and eventually had a huge launch of the product, publicly stating that they were sunsetting their legacy application; and it fell flat. The new rewritten application was inefficient, severely lacking in functionality and suffered from a disconnect between the offshore team and the local team. They lost some customers and the ones they kept stayed with the original product. It was a costly mistake that they didn’t want to make again.

They realized that they needed to modernize without having to start from scratch, keeping the COBOL where it was strong - processing data, and using other languages where COBOL was lacking - the user interface.

Highlights

- Preserved valuable application assets and irreplaceable programming skills
- Enabled organization to take advantage of modern Java technology
- Retained long term customers and gained new sales opportunities
- Provided Web enablement, SOA, and Web Services
- Offered enhanced graphical user interface (GUI) capabilities

THE DECISION PROCESS

Partner Jim McMillian came to Matthew one day and said “This seems too good to be true, but I was looking for a modern COBOL platform that would essentially allow us to modernize the application with relatively low effort, and out came Veryant.” As they evaluated isCOBOL they were always waiting for the gotcha moment; the “ Oh, this is why it won’t work” revelation. But they never found the gotcha moment, instead discovering isCOBOL to be everything it promised to be. What helped them make the decision to use isCOBOL was the personal connection. For instance, during the evaluation phase, Veryant’s sales director and Veryant’s CIO, Massimo Bertoli came to their office in Alabama to meet them, give them a presentation, and answer their questions.

THE RESULT

Collection Partner heavily relies on Veryant’s EIS (Extend Internet Support) capabilities. isCOBOL is their API back end, and they rewrote their user interface using React JavaScript and the Electron framework. This combination allowed them to develop a modern front end, while keeping the COBOL business logic and accounting functionality, which Matthew calls “The main power in our software”.

To keep their COBOL, Collection Partner had to “decouple” their user interface from the business logic and create separate COBOL programs, each acting as a REST API, accepting and returning data to their user interface. There were some challenges with the mature, “spaghetti” code, with GoTo’s and intertwined UI. But about 90% of the decoupling was pretty straightforward and involved removing the user interface ACCEPTs and validations. In order to create the REST services, Collection Partner uses isCOBOL EIS Service Bridge. The creation of the REST Web Services is automatically done by the isCOBOL Service Bridge facility. With this feature enabled, every time the Compiler compiles a legacy COBOL program with a Linkage Section, it generates a bridge class that allows the program to be used as a Web Service. The process was made even easier by using isCOBOL’s Eclipse-based IDE, which automates the process further and makes testing easy.

Collection Partner feels like they’re working with Veryant in a team effort, with “outstanding” customer support and responsive developers. They also appreciate the fact that Veryant is constantly making improvements and enhancements to their products and is very responsive to customer’s needs, with a support mentality of “Let’s take a look at where we can better our product and documentation.”

Using isCOBOL’s compiler, runtime, IDE, and EIS WebServices Service bridge, Collection Partner has created a very stable, low maintenance, back end environment coupled with a cutting edge, modern user interface.

“We wouldn’t be able to do what we’re doing without Veryant’s EIS functionality, it’s the whole reason that this project is possible.”

- Matthew Burke, CTO of Collection Partner

Corporate Headquarters

6390 Greenwich Dr. - Suite 225
 San Diego, CA 92122 - USA
 Tel: +1 619 453 0914
 Fax: +1 858 569 4243
 info@veryant.com

European Headquarters

Via Pirandello, 29
 29121 - Piacenza - Italy
 Tel: +39 0523 490770
 Fax: +39 0523 480784
 emea@veryant.com