



Analyst Report: Value-Based Case Study

Managing an IBM MQ Environment at USBank



Originally published in

C3B Consulting's

INSIGHT-SPECTRA

Applying common sense to using technology intelligently

MANAGEMENT INTRODUCTION

USBANK is headquartered in Minneapolis and is #6 in the US (as measured by assets), USBank has some 50,000 employees. **David Corbett** is a middleware architect in the Bank. He focuses on multiple forms of middleware: this includes CICS and IBM MQ plus TIBCO messaging (USBank, in common with most large organizations has no shortage of different forms of middleware installed).

In this case study, Mr. Corbett discusses the development and operational issues in USBank that arise with IBM MQ and how these are managed. He describes issues arising between operations and development and explains how and why USBank introduced Avada Software's Infrared360 solution to improve both the Bank's application development and operational environments.

THE BUSINESS PROBLEM

We do not have at USBank, a complex IBM MQ environment (I know of ones that are much more complex). Our IBM MQ network supports traffic to and from the mainframes but there is also traffic over the IBM MQ network that goes between non-mainframe (distributed) systems. IBM MQ is, therefore, an integral part of how we make applications work together within the Bank. As such we need access, for all sorts of valid reasons, to see what is happening in the queuing infrastructure — but with strict controls operating on what can be seen by whom (for security and confidentiality reasons).

A further consideration is that we are a bank. We are highly regulated (by the Office for the Control of the Currency) and there is always great concern about traceability and logging. It is vital, both for our business integrity as well as to satisfy regulatory and compliance requirements, that we know — and can show — what goes on where, why, and how.

This has some interesting implications. Take, for example, the concept of most middleware tools, like MQ, WAS, etc. They 'demand' a customer has a designated administrator, who will have the necessary privileges to operate that software and, with these privileges, complete control over sensitive transactional and core banking systems. Banks struggle with this because those that know and own

expertise about how to use a particular tool in-depth are often in development roles. But developers generally (and correctly) should not have the visibility and access rights that (say) someone in operations must possess.

In effect there can be a conflict of interest here. This needs to be managed most carefully so regulatory and compliance requirements are fully satisfied. To accomplish that, there must be evidence that the requirements are satisfied. For example, at USBank we have had to create emergency access IDs for certain types of operations; these are held by security. If we, in development, are asked to solve a problem which we (in development) would not normally be allowed to do, we must request an emergency ID from security and the subsequent actions using this ID are tightly monitored and logged.

In addition, a background factor was driving us to look for new approaches. This was our need to reconfigure our IBM MQ network which had evolved over time. As we grew, we did not have what I might call an 'MQ Farm' with multiple applications pointing to the same place, but rather we tended to have several implementations where there was a dedicated IBM MQ Manager for nearly every application using IBM MQ. This was expensive and not optimally efficient. IBM MQ was an obvious candidate for reevaluation and restructuring, not only to reduce costs but also to make its usage more coherent and manageable as we grew further. As we added more MQ servers, to improve the structuring of our MQ environment, it became even more important to have a modern management tool.

DEVELOPMENT AND PRODUCTION IN AN IBM MQ WORLD

IBM MQ is an interesting piece of middleware. Normally, in most production environments, Operations can run with what works. This is not so simple with IBM MQ, which often requires Operations teams have a good understanding of the related and associated applications in their MQ environment. This means that application development often becomes an integral part of ensuring an IBM MQ network continues to function. On this basis, IBM MQ can be regarded as something rather different to most middleware.

Yet it is difficult in a large organization to step up to the plate to take ownership of how IBM MQ works across the enterprise. Putting everything together for Operations is

dependent on how much Operations is prepared to take ownership of IBM MQ. Part of the issue is where IBM MQ is running. For example, our mainframe IBM MQ implementations — and, for that matter our HP Nonstop systems — function in a normal, big computing environment: this is proven. But our instances of IBM MQ running on our iSeries, AIX, HP-UX and Windows environments often do not share the same operational disciplines as we have for our mainframe/Nonstop environments — even though they do ‘talk’ to one or the other (or both) of these larger systems.

What many people do not wholly understand is that IBM MQ, once it is running, is extremely stable. But one (unintended) consequence is that, once running and stable, people forget what they have done and why. It may seem strange but one area where Avada Software's Infrared360 assists us is with ‘coping’ with difficulties that arise when something breaks (which everything does at some point) and when institutional memory cannot recall what should happen. Avada's software gives USBank that extra degree of depth and understanding that overcomes the difficulties which arise when people cannot remember.

“You can’t know if [a SOAP-based] service is working as expected just because the service is responding; the response may be invalid and hide issues with the services.”

It institutionalizes knowledge about key areas of our middleware environment.

WHAT WENT BEFORE AND WHAT WE NEEDED

Before we purchased the Infrared360 product from Avada Software (Avada) we had a tool for developers with which a developer could go and look at his or her queues (and only their queues) to see what was happening. It could also do some limited manipulation, but this was not sufficient for us.

Why was this important and even a business issue? IBM MQ is middleware that is essentially invisible but is also tightly integrated to the applications which it connects via its messages and queues. As we have found out the hard

way, it is often very difficult for production/operations staff to understand how an IBM MQ network runs or what are the application implications. In addition, production did not possess any tools of its own to make this easier.



The result was that, when a production problem occurred that was IBM MQ-related, the operations people all too often could not resolve whatever was wrong. Its solution was to call development to seek assistance in fixing the problem. In essence what we were looking for was a solution which could:

- Monitor our IBM MQ network
- Enable those who had to manage production to do this without needing specific IBM MQ expertise
- Avoid the need for installing agents
- Authenticate users via our LDAP tree (for security, so that we could know actions were only performed by those authorized)
- Be browser-based (which was very important to me personally — because it is a commonly understood user interface and avoids having to install clients everywhere)
- Reduce the load, thereby, on development
- Support our restructuring of the MQ environment with our introduction and management of a farm of IBM MQ servers.

INTRODUCING AVADA'S INFRARED360

We heard about Avada almost by accident, from our Tandem technical services support person. It then emerged that I knew one of the founders and developers from when he worked at a previous software company. As we were

already looking at other software (specifically MQSoftware and BMC's Patrol), we included Avada in the evaluation we had started.

In the assessment what became clear was that, although the solutions from MQSoftware and from BMC had good features, their biggest drawback was that both would require us to run agents everywhere (which would add a complexity we did not want). In addition, both seemed to demand we have somebody almost full time just to manage their tools. This did not make much sense to us; nor was it attractive.



A further factor pushing us towards Avada was that its solution was less expensive. But most of all Avada appeared to meet more of the requirements that I described above than its competitors. The final 'plus' was it offered a browser interface (removing that need for clients) — which was unique amongst these three possible solutions. Indeed, if I look back, LDAP integration for security was a key consideration. While all three offered this, only Avada's Infrared360 capitalized on that with the ability to permit or limit access based on users' permissions.

AVADA'S SOFTWARE IN PRACTICE

The browser-based interface is as we want, though at first it was not as intuitive as I would have liked. Over the years, subsequent versions have improved on that greatly, and initially we were able to address this with user training. Once users are trained, it works and user friendliness certainly comes with greater familiarity.

What we have found is that the Avada approach does enable us to be more flexible. In fact, Infrared360 has a unique ability to string rules together for more granularity and more powerful rules, search, and filtering. We can write flexible rules, of increasing complexity, which satisfy our needs. (This is one of those areas where it is often not easy to be sure you have bought correctly- until you are

deep into using a tool. The evidence has proven we made the right decision.)

Rules in Avada are XML-based. This has various advantages. We understand XML. This means that, if we need to, we can go in and tweak the rules through editing the XML — which is relatively painless.

When it comes to tying various aspects together, Avada has deep capabilities. For example, if a channel is not running due to a failure on the sender or the receiver side, and therefore the outbound queues have undelivered messages. Then, we find those messages have been there for 10 minutes and no one is reading or writing to that queue. Addressing this sort of problem in an automated way is not easy at any time. But Avada makes it much easier in such 'compound' situations. Infrared360 has automated pulldown menus to address this, so there is no need to bring development in to write scriptable logic to manage all this. Over time we have used more and more of these sorts of capabilities.

Indeed, over the years, we have not come up with any rules scenarios that Infrared360 has not been able to support. Equally, the capability to build a rule and re-use it (without having to redevelop it) has been a big time saver.

“One area where Avada Software's Infrared360 assists us is with ‘coping’ with difficulties that arise when something breaks... Avada's software gives USBank that extra degree of depth and understanding that overcomes the difficulties...”

Besides helping us solve the issues outlined above, Infrared360 possesses a statistical database (operating behind the scenes). We saw a lot of possibilities for exploiting the statistics that are generated. We've used this for analysis, for identifying weak points or bottlenecks, etc. But one particular advantage is that we can tie it to services. So, we've set up services that help us speed time to resolution for situations that can't be avoided through a proactive management approach. For example, there may be a queue with messages coming from an outside vendor

with very low volume (a few messages per hour or less — but critical). Should there be a situation where the vendor says a transaction was sent that weren't processed, we are able to use this feature to confirm that the messages never were sent to us. Infrared360 makes it easy to set up the service, easy to run, and easy to look at the resulting analytics.

Fundamentally, however, what we have built with Avada is an IBM MQ environment where the Bank obtains preemptive notification of problems. This proactive monitoring and management approach is quite different to the reactive environment with which we had to work in previously — and much sounder as well as more reliable. Now we do not have to wait for people in the Bank to call us to say 'we are not receiving our messages' as the first indication of a problem. We are fixing the problem before they can even call us — because we already know.

The net result is that application development, where I am a middleware architect, provides operational support to production for non-mainframe/Tandem functioning of IBM MQ. To do this Avada's Infrared360 product has become significant in our effectiveness — so that the development staff can see their queues on mainframes as well as distributed systems. This is not so much from the monitoring perspective as the ability to see what is happening when the problems occur.

LESSONS LEARNED AND BEST PRACTICES

As a regulated financial institution, we had and have particular security requirements. I would say from our own experience that even if you do not have such obligations, it is both desirable and a good practice to sit down and formally lay out the groups and roles that you want to assign — before you start building- out your application accessibility.

Equally you should spend time working through what are the generic rules you should apply, and then create these. Doing this early will save much time later. Even though this did take us longer than I expected to come up with an acceptable model for the groups it was part of learning to use our new tool.

For controlling access in the way we wanted it was necessary to do the filtering at the group level, which was pretty much self-evident. Defining the groups — as well as which IBM MQ queue managers and what could be seen — was trickier. It did force us to think clearly, which was particularly necessary for system queues. Now done, the time invested was worth it.

One other indicator of success is worth mentioning. Our security, developers, and operations are much happier than before. Avada's software enables us to satisfy their needs in ways that do not hinder — and even assist developers in doing their jobs. Because of the uniqueness of Infrared360's design, its Trusted Spaces™ feature, and its thorough user and historical change logs, other solutions would not have had such success in this area.

MANAGEMENT CONCLUSION

USBank is typical of many large organizations - with a disparate middleware infrastructure. As Mr. Corbett illustrates, where USBank is different is that it has looked at its IBM MQ environment from a modern banking and compliance perspective and sought to consolidate and improve.

The key lies with how IBM MQ works in practice in most organizations. Broadly it is so reliable it is forgotten about. In one sense this is good — not much support energy has to be expended. But in other ways this leads to downstream problems — like in how development and operations should work together.

In selecting Avada Software's Infrared360 product, USBank introduced a flexible solution that helped reduce costs by reducing the workload on developers working with IBM MQ, enabling Operations teams (leading to reduced down times and time to resolution), and facilitating the restructuring of the MQ environment. All while meeting the internal security requirements.



**FOR MORE INFORMATION ON AVADA'S INFRARED360
OR TO ARRANGE A DEMO [CLICK HERE](#)**

100 Enterprise Drive, Ste 301 Rockaway, NJ 07866 +1 (862) 781-6159