

## Creation of a Risk Data Application

**Client:** Anonymous

**Business Size:** Global Corporation

**Industry:** Commodities Trader

**Country:** International

**Technology:** C#, Python, .NET Framework, LINQ, Confluence

### Objective: Install and Configure Upgraded Risk Management System

#### The Brief

The project was to setup a Windows Server 2019 platform, run as a VM, install ODBC drivers, install and configure Octopus Deployment Agent, ActiveBatch Execution Agent, implement server security, and install and configure the very latest version of RiskManager. All this had to operate within the client's security policies.

#### The Client

The client is a multinational commodity trading and mining company. They use sophisticated risk management software and systems to assess and mitigate the various risks associated with trading commodities. These software solutions are designed to handle complex calculations, analyse market data, and provide insights into potential risks.

Within their Oil and Gas commodity trading department, a third-party software system called RiskManager, supplied by MSCI, is used to calculate the risk involved when trading.

The client operates in numerous segments of the commodity industry, including:

- Commodity Trading, as a major player in the global commodity trading market. The company engages in the buying and selling of a wide range of commodities, including metals, minerals, energy products, and agricultural goods. It acts as an intermediary between producers and consumers, facilitating the movement of commodities across the globe.
- Mining, involved in the extraction and of various natural resources. They own and operate mines that produce commodities such as copper, zinc, nickel, cobalt, coal.
- Agriculture, trading and marketing agricultural products such as grains, oilseeds, sugar, and cotton. The company is involved in the supply chain from production to distribution.

## Case Study

### *International Commodities Trader*

- Oil and Gas, engaging in the exploration, production, and trading of oil, gas and petroleum-based products.
- Logistics and Storage, managing logistics and storage facilities to support its trading and mining operations. This includes transportation, storage, and processing infrastructure for various commodities.

The client's business is highly diversified, with a global footprint with operations in multiple countries. Its activities span the entire commodity value chain, from extraction and production to trading and marketing. Their business model involves managing and optimizing the supply chain for a wide range of commodities, contributing to its status as one of the leading players in the global commodity market.

### **Challenges**

The main challenge was to install and configure the latest version of RiskManager. The primary reason was the current version that had been previously installed in 2018 was running on Windows Server 2012, which no longer satisfied the client's security policies.

Another challenge was understanding how RiskManager actually worked, its dependencies, the type of framework it required and the type of webserver it required. Furthermore, it had not yet been tested on Windows Server 2019.

There was limited documentation regarding the peripheral processes which were responsible for uploading data to RiskManager, and so an investigative approach was required to understand in as much detail as possible the composition of the entire system.

The greatest understanding was achieved by reading what documentation did exist with the client and that provided by MSCI, going through each job within RiskMetrics ActiveBatch execution, and talking to key members within the RiskMetrics department who had knowledge of the peripheral systems and peripheral data. All this combined, allowed a reasonable view and understanding of RiskMetrics, and the peripheral systems with data that supported it.

The installation process of RiskManager was largely manual, because although there was an installer provided by MSCI, it simply created a folder structure. DLLs had to be manually copied to the bin folder within the root of the application, and XML files had to be copied and correctly configured. Furthermore, every server required the ActiveBatch execution agent, the Octopus deployment agent, and SAP IQ Client Suite 16.0 mainline ODBC drivers to connect to a Sybase database and much more.

Once the system had been set up, a market data download was necessary. It was a colossal dataset, taking nearly 5 hours to download. Sometimes it would contain invalid data, which would stop the entire system from functioning. This rendered it

## Case Study

### *International Commodities Trader*

necessary to trawl through log files to understand which part of the system was failing. It was also necessary to liaise with an MSCI representative and troubleshoot the system in an attempt to fix it. In many cases, due to the intermittent nature of this problem, the next download would be successful.

### **Methodology**

With the complex set of installations required, a step-by-step testing approach was adopted. To deliver the entire RiskMetrics system it was necessary to install, configure, and then test the following: OpenJDK, Apache Tomcat webserver, RiskServer, ActiveBatch execution agent, Octopus deployment agent, SAP IQ Client Suite 16.0 mainline ODBC drivers, RMClient. At every step of the installation process, the system configuration was tested, with an additional test report generated after data upload. In parallel, documentation including detailing system components, troubleshooting, file locations, and how to debug a data upload was created in Confluence.

The development team used a variation of the Agile methodology which worked very well with a daily morning stand-up, sprint planning, and retrospectives. Team members could collaborate as and when necessary, and Microsoft Teams played an essential role in communication with various departments which were responsible for different aspects of the system (such as VM creation and configuration, firewall configuration etc).

### **Results**

The validity of the data was of utmost importance. To assist in verifying the behaviour of the new systems, the old and new servers were run in parallel for a number of weeks, making daily comparisons of the datasets, verifying that they were the same, before switching over to the new server and moving that server to production.

The project was a success despite the many potential pitfalls and the client was able to continue to use RiskServer within their security regime.

### **Skills and Expertise**

- C# and Python Languages: Including Object-Oriented Design with Classes and Objects, Inheritance and Polymorphism, Exception Handling, Collections.
- .NET Framework: Understanding the basics of the .NET framework, its architecture, and the Common Language Runtime (CLR).
- Visual Studio: Proficiency in using Visual Studio for coding, debugging, and building C# applications.
- Console Input/Output
- Error Handling: Ability to handle errors and exceptions appropriately.
- LINQ (Language Integrated Query): Basic understanding of LINQ for querying and manipulating data collections.
- Debugging: Proficiency in debugging code.

**Case Study**  
***International Commodities Trader***

- **Version Control:** Understanding of version control systems (Git) for source code management.
- **Testing:** Knowledge of unit testing frameworks (MSTest, NUnit) and the ability to write and run effective code tests.
- **Documentation:** Good documentation practices to make your code understandable to others.