SCS Case Study

Global Investment Bank Trading Application upgrade

The Client

The client is a leading investment banking firm with headquarters in downtown Minneapolis and several branch locations in the U.S., London, Zurich and Hong Kong.

The Challenge

Prior to Superior Consulting Services' (SCS) involvement, the client migrated several applications from .NET 3.5 to .NET 4.0, but they were never implemented to production environment due to several challenging issues. The Technical Research team uses these applications to analyze tickers/stock, symbols/exchange symbols, and produces analysis reports for their customers. This project consists of different types of applications. There is an ETL process that runs on a daily and weekly basis to export and send data to customers. Several console applications are scheduled to run a portion of the day to get the most recent stock pricing information via API. There is also a Windows application that is installed on each Technical Researcher's machine, and there is a Windows service application known as the base application, which runs 24/7 on a server listening for all requests coming from the rest of the applications. The communication between applications is through TCP/IP with a specific port number. All .NET applications were written in C#. There are also third party tools being used, such as Crystal Report for reports, Nevron .NET for charting, and ABCpdf .Net for PDF stitching.

An existing memory leak problem had also been discovered prior to .NET 4.0 migrations. When this happens, the service crashes at midnight and may result in missing PDF reports. This requires rerunning the service and manually reprocessing the PDF reports the next day. If, for some reason, the support and maintenance is not available before business hours, this slows down the business process and causes delay to the Technical Research team.

Additional issues were identified after migrating to .NET 4.0. The logging feature and a few custom libraries in the Windows service application suddenly stopped working in Windows 7. The existing memory leak problem in .NET 3.5 got worse. In addition, the Crystal engine was exceeding the process limit when processing a large number of Crystal Reports during weekend report generation. All these issues were identified in the pre-production environment. With the company's goal to convert all applications to .NET 4.0, it was critical to resolve all these issues.

The Solution

SCS was contacted to help investigate and resolve these issues. An SCS resource was deployed onsite to work with the client's IT team. The first course of action taken was to reproduce the known issues. The biggest challenge was to capture the right sequence and timing to trigger these errors since most of them were intermittent cases. We worked with the project owner to verify the issues and gather the facts. Analysis of the issues required collaboration among SCS experts, the client's R&D team and one of the software vendors. The next move was listing all the potential causes and appropriate solutions to the issues. We spend time to verify each solution and performed all testing. In the end, all issues were resolved and we promoted this application in NET 4.0 to production. It is now running and very stable.

The memory leak problem was resolved by hunting down the unused/live objects and disposing them properly right after utilization. This fix also improve the application performance by replacing the old third party PDF with the newer tool. The old tool was using a lot of memory resource when processing large numbers of reports. The new tool discovered how to optionally limit the memory usage and resulted in better performance.

The logging functionality of the client application in.NET 4.0 suddenly stopped when deployed to Windows 7. This was simply resolved by setting the application to run as administrator on the machine. The code writes information to the Windows event log. Windows 7 has additional security that needs to be configured for applications accessing the Windows administration components like event logs, etc. Another issue that emerged was the incorrect bit-ness setup of the report project or custom library. This library contains Crystal report 32-bit assemblies but is configured to use Any CPU. By default, Windows 7 64-bit automatically runs this library as 64-bit when configured to Any CPU. This was fixed by configuring the library to run as x86 to match the Crystal Report 32-bit assemblies.

In addition to setting the client application to run as Administrator and configuring the report project to use the correct bit-ness, we also enhanced the reflection code and replaced the old class object type used to instantiate the Crystal Report object with the dynamic type in .NET 4.0. It turned out that the old object type was not really disposing the Crystal Report object even though it appeared to be using the Dispose method. After implementing the Dynamic type, this error was resolved as well.

The Result

Currently, the processes in production have been very stable. The project was migrated to .NET 4 without too much maintenance and troubleshooting. The Technical Research team can also use their time appropriately without waiting for something to be fixed first thing in the morning. The client is very happy with the end result, the timeframe of the investigation and that the appropriate solutions were applied to the problems. The client chose to keep the SCS consultant involved in support and maintenance for this project, and even assigned new projects in 2013 and 2014.

Contact

Superior Consulting Services, LLC
350 West Burnsville Parkway, Suite 550
Burnsville, MN 55337
952-890-0606
www.teamscs.com