



Case Study:  
Spread for Windows Forms

Robust Spreadsheet Capability for  
State Agency Application

## Contents

Introduction.....	2
Background.....	2
Original Application Design.....	4
A Better Solution.....	5
Conclusions.....	11
About HIS Consulting Services.....	11
About the Author.....	12
More Information.....	12

The FarPoint logo, featuring the word "FarPoint" in a bold, black, sans-serif font. The letter "o" in "Point" is replaced by a blue sphere with a white highlight, giving it a 3D effect.

Enterprise Spreadsheet Solutions  
[www.FarPointSpread.com](http://www.FarPointSpread.com)

## Introduction

“With ... FarPoint Spread, we have only scratched the surface of what is one of the most impressive, flexible, and capable components I have come across.”

R. Kevin Chandos  
Senior Technical Consultant,  
HIS Consulting Services

Desktop applications are becoming more robust and the State of Washington Department of Labor and Industries (L&I) has very intense data crunching demands. Along with the requirements for user interface ease, the agency needs back-end calculation power for its applications. L&I requires reporting a growing amount of data that must be easily managed and read by several personnel in several sections of the department.

Working with Excel files, providing an easy interface to learn and use, and handling large amounts of data quickly are only some of the requirements that the project team faced in application development.

The team found a solution that provides incredible spreadsheet capability. The solution is more robust and customizable than Excel, but remains familiar to users. The tool integrates seamlessly with L&I's Visual Studio development environment. The engine behind much of the power of our data analysis is the spreadsheet capability of Spread for Windows Forms from FarPoint Technologies.

## Background

The State of Washington Department of Labor and Industries (L&I) is a diverse State agency dedicated to the safety, health and security of Washington's 3.2 million workers. L&I helps employers meet safety and health standards by inspecting workplaces when alerted to hazards. As administrators of the State's workers' compensation system, L&I is similar to a large insurance company, providing medical and limited wage-replacement coverage to workers who suffer job-related injuries and illness. L&I's rules and enforcement programs also help ensure workers are paid what they are owed, that children's and teens' work hours are limited, and that consumers are protected from unsound building practices.

Employers in the State of Washington must pay industrial insurance premiums to L&I based on the type of work performed and the total number of hours of that work performed by their employees. L&I has a Field Audit section that performs audits of the premiums paid and hours reported to L&I and has the authority to levy fines. The Field Audit section is also responsible for performing audits and collecting fines against business who fail to register or otherwise operate illegally in the State of Washington.



The agency goal, approved by the State Legislature, was to replace the existing system – a collection of non-integrated application and utilities some of which are over ten years old. The name for the application is FACT - Field Audit Computer Technology. This application supports field audit personnel in the collection, collation, reporting, and dissemination of data in support of audit activities. It pulls together information from servers and mainframes from both within the Department itself and from other State of Washington agencies, thus providing auditors with a complete picture of what a business has reported (or failed to report) to the appropriate State agencies. The application allows auditors to add data based on the employers' own records, timecards, checkbook, etc. From this data, fairly complete assessments can be made. In addition, this application would support the legal concerns of the department by providing litigation audit summary information and audit tracking.

## Original Application Design

The first version of the application was written in Microsoft Visual Basic .NET (2005) and consisted of many parts that included two front ends (one Windows Forms, the other browser based), a back-end data assimilation engine (which packages data for the front ends), and a back-end data integration engine (which provides access to data without the front end knowing or caring whether the data came from a local SQL server, a network SQL server, a mainframe, message queues, or a Web-service).

The primary data entry mechanism for the application was Microsoft Excel operating as a hosted component. It should be noted that Microsoft does not recommend this method. Microsoft Consulting Services was brought in and recommended an unsupported C++ component (DSOFramer) to aid in the hosting.

The application was released to production, but the Windows Forms front end did not perform adequately. Excel did not function as expected in the hosted environment. Events were not propagated to the hosting application, recalculations were not properly triggering, and the performance was very slow compared to required levels. For example, a typical first-cut audit, which included very little data, was taking nearly 40 seconds to load. This was a limitation that required a different solution. The ability to crunch data was the primary purpose of the application as a whole.

HIS Consulting Services was contracted to investigate the performance and user interface interaction issues and assist in the development of a solution. Beyond simply getting something that could crunch data, we strove to make the application work so that the users would accept it and would be productive with it as well. This was not an easy task – we were dealing with over 200 users whose primary concern was the collection of data from employers, not the nuts and bolts of storing the data. They were accustomed to storing the data disconnected and in free-form via Excel, and had never had an integrated application to support their end-to-end business process.

## A Better Solution



From the start of this next phase, the project team determined that the Excel integration must be replaced with a more robust solution. The existing .NET GridView control would not perform as needed without an extreme amount of sub-classing and rebuilding. This determination is what started us down the road to finding a third-party component. At the minimum, the component had to be able to read and write Excel files, support formulas, support multiple cell types, provide for customizations such as colorization, font, cell spanning, and be able to expose data change events to the hosting class without clumsy work-arounds and interop overrides. In addition, the component had to be easy to learn because it had to be implemented very quickly once it was acquired. L&I has very strict requirements regarding what can be assimilated into its application infrastructure. Based on these requirements, we narrowed down the research area to four leading spreadsheet controls on the market. After a two-week evaluation period, it was clear the FarPoint Spread for Windows Forms was the hands-down winner. No one else came close to the ease of use, ease of implementation, cost, flexibility, documentation and Web based support for the control.

The next thing to be done was to remove Excel from the application and create a new View class that contained multiple FarPoint Spread controls. The removal of Excel took approximately three days, while the initial "rough-in" of the FarPoint Spread took an additional two days. The initial development included the control layout, activation, event hooking, and preliminary colorization. Once the rough-in was done, we fired up the application just to see what kind of loading performance we might expect. We were more than a little surprised to see a typical 40-second load time drop to under 4 seconds. Further, we loaded one of our largest audits which was averaging about 90+ seconds load time, and we were delighted to see it load in less than 10 seconds. This told us that the largest overhead in the control was our custom initialization, not the loading of data.

Functional modifications that used to take a day or more now took only a few minutes to code and a few more to test. Since FarPoint Spread is a true .NET component, there were no "unexpected" interactions, unknown events, or failed event firings.

Each page of the user interface is implemented by an instance (sheet) of Spread. Screen shots of some of the sheets that make up the user interface are shown below.

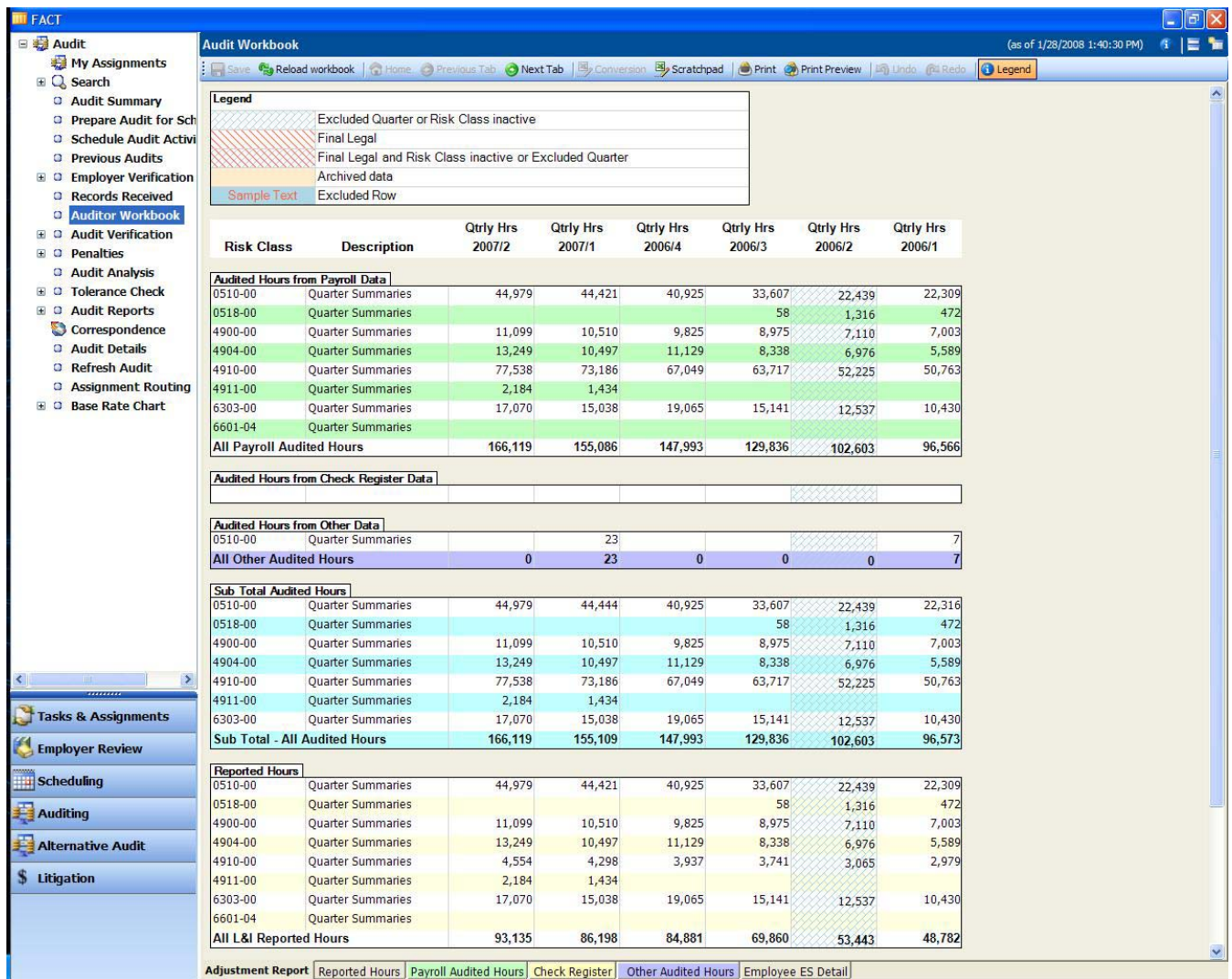


Figure 1. Adjustment Report Sheet of the Audit Workbook

Three sheets are used for data entry, three for summarization of entered data, six for summary assessment, one for legend display, two for titling, and one for report preview packaging. The three data entry sheets incorporate computations as well as event processing for database look-ups based on multiple-cell contents, while the nine summary sheets are all tied together through event processing on any data entry change.

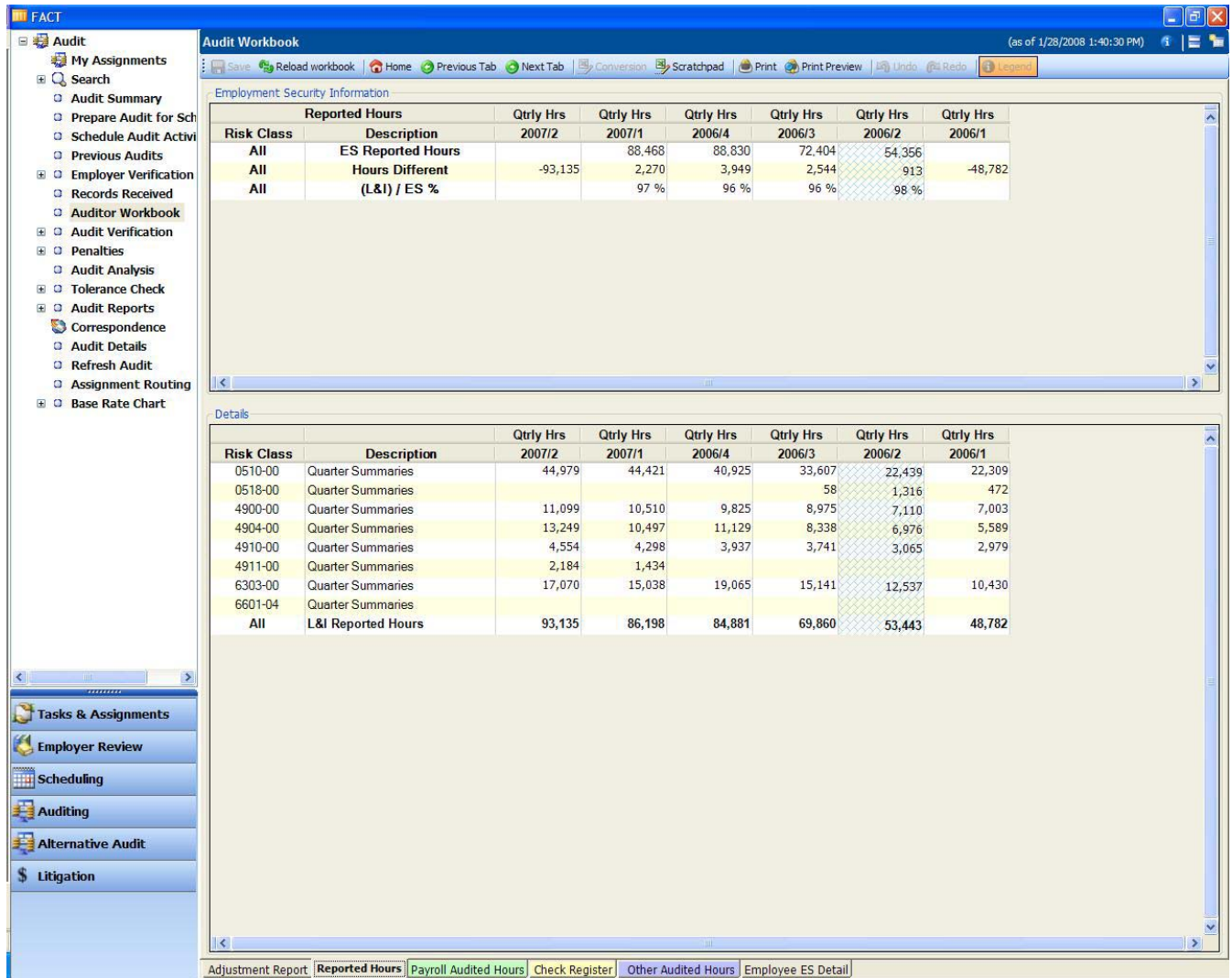


Figure 2. Reported Hours Sheet of the Audit Workbook

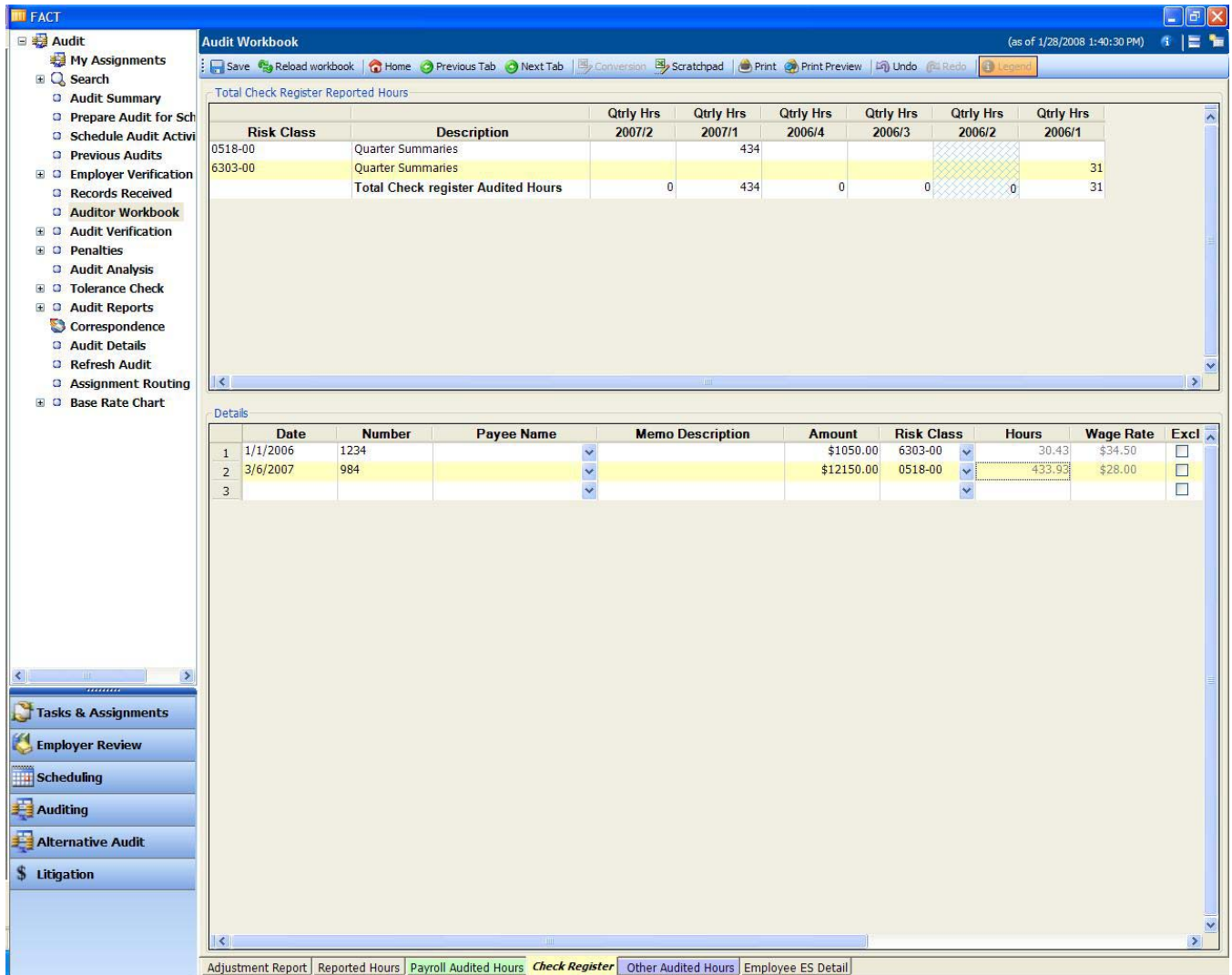


Figure 3. Check Register Sheet of the Audit Workbook

Cut and paste (Ctrl+C, Ctrl+V), calendar pop-up, and calculator pop-up are supported intrinsically in the control saving on code production and both development and testing time. Drag-and-drop from external sources is supported with specialized drop handlers in order to pre-validate data before incorporation into each data entry sheet. To speed the drag-and-drop processing, event handlers are disabled during the drop operation and re-enabled only after the data has been validated, applied, and summarized.

During testing a block of 1,400 rows of data was dropped on the "Other Audited Hours" sheet, and the entire drop, verify, incorporate, and summarize process took less than one second! Have we mentioned how fast this control is?

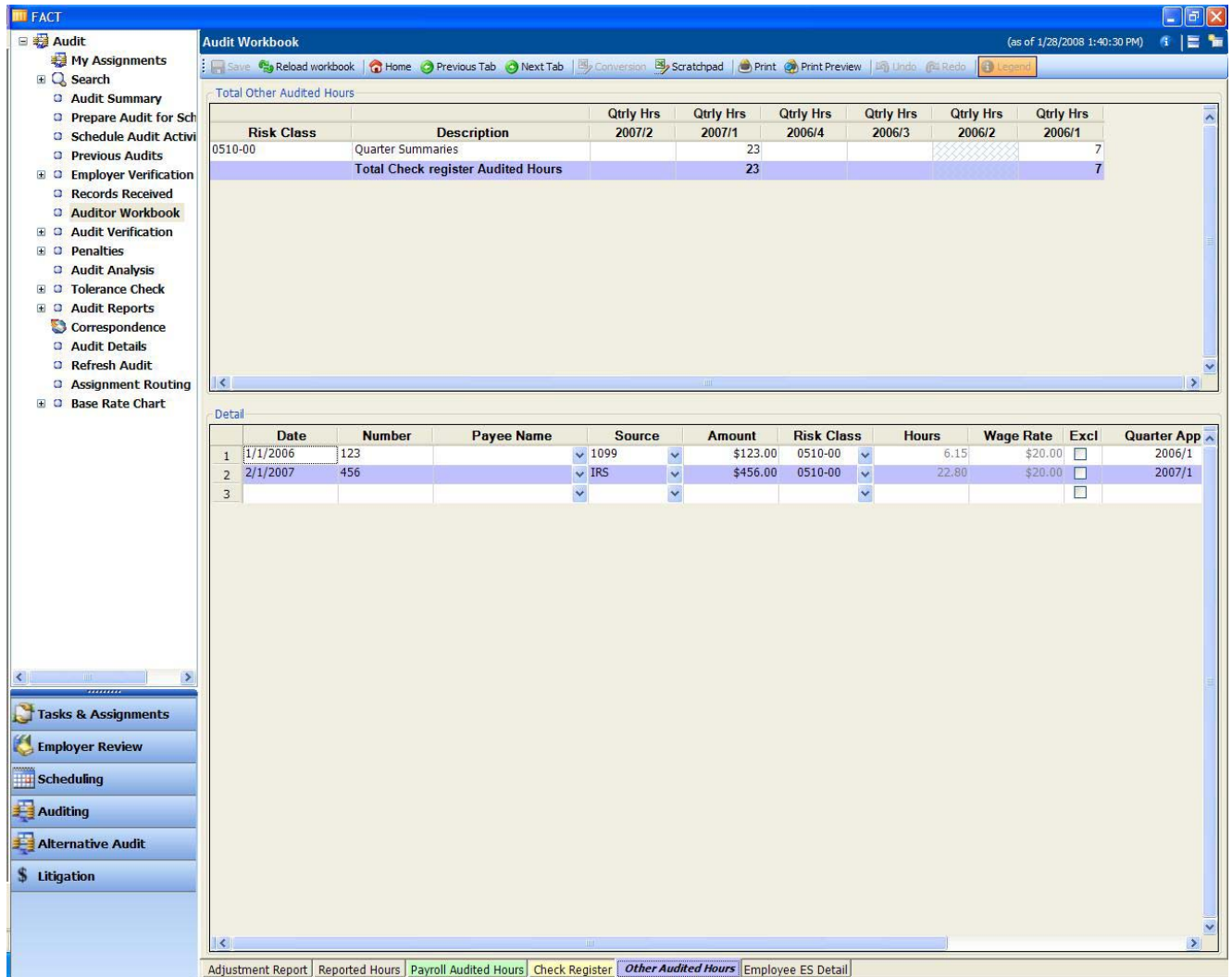


Figure 4. Other Audited Hours Sheet of the Audit Workbook

Print preview

Summary Adjustment Sheet - Page 1 of 2

Risk Class	Description	Qtrly Hrs 2007/2	Qtrly Hrs 2007/1	Qtrly Hrs 2006/4	Qtrly Hrs 2006/3	Qtrly Hrs 2006/2	Qtrly Hrs 2006/1
Audited Hours from Payroll Data							
0510-00	Quarter Summaries	44,979	44,421	40,925	33,607	22,439	22,309
0518-00	Quarter Summaries				58	1,316	472
4900-00	Quarter Summaries	11,099	10,510	9,825	8,975	7,110	7,003
4904-00	Quarter Summaries	13,249	10,497	11,129	8,338	6,976	5,589
4910-00	Quarter Summaries	77,538	73,186	67,049	63,717	52,225	50,763
4911-00	Quarter Summaries	2,184	1,434				
6303-00	Quarter Summaries	17,070	15,038	19,065	15,141	12,537	10,430
6601-04	Quarter Summaries						
All Payroll Audited Hours		166,119	155,086	147,993	129,836	102,603	96,566
Audited Hours from Check Register Data							
0518-00	Quarter Summaries		434				
6303-00	Quarter Summaries						31
All Check Register Audited Hours		0	434	0	0	0	31
Audited Hours from Other Data							
0510-00	Quarter Summaries		23				7
All Other Audited Hours		0	23	0	0	0	7
Sub Total Audited Hours							
0510-00	Quarter Summaries	44,979	44,444	40,925	33,607	22,439	22,316
0518-00	Quarter Summaries		434		58	1,316	472
4900-00	Quarter Summaries	11,099	10,510	9,825	8,975	7,110	7,003
4904-00	Quarter Summaries	13,249	10,497	11,129	8,338	6,976	5,589
4910-00	Quarter Summaries	77,538	73,186	67,049	63,717	52,225	50,763

Figure 5. Print Preview of Summary Adjustment Sheet

All of this was accomplished while protecting data integrity, minimizing application modifications, and responding to (sometimes aggravating) user requests, in less than 50 work days. Our production roll-out was a raving success and moved the application from the cutting-block stack to the "here's a fine addition to our portfolio" stack.

## Conclusions

FarPoint Spread saved the agency extensive hours in on-going development and maintenance, as well as shielding the staff from potential and unknown incompatibilities when Microsoft Office 2007 would be rolled out to all department machines. Microsoft gave no indication that hosting Excel would work any better with the 2007 version, and no guarantees that Excel 2007 would function anything like Excel 2003 as a hosted component. Between the upcoming Office 2007 deployment and the pending Microsoft Windows Vista deployment within the agency, the use of FarPoint Spread has successfully mitigated the risk of a work stoppage for the agency audit staff, while empowering the auditors, and changing conversations from "*... and this just doesn't work ...*" to "*... this is great. Now we want to add ...*". The choice of Spread fulfilled our performance requirements and accelerated application development.

This success is being propagated as the agency is now examining FarPoint Spread for ASP.NET, the Web Forms component, for another project with a public-facing data-acquisition requirement.

On a final note we would like to point out that we are not using anywhere near all of the capabilities of FarPoint Spread. With our new application incorporating Spread, we have only scratched the surface of what is one of the most impressive, flexible, and capable components I have come across.

## About HIS Consulting Services

Since 1989, HIS Consulting Services has been providing data systems that integrate mainframe systems and application-server based systems, primarily for state government. With a staff of nine, we have earned a reputation for delivering high-performing systems on-time and on-budget. Kevin Chandos is an excellent example of the expertise that we proudly offer to our customers.

## About the Author

**R. Kevin Chandos** is currently a Senior Technical Consultant with HIS Consulting Services in Olympia, Washington, and his programming experience includes accounting, scientific, military, insurance, engineering, and health care industries and encompasses complete development from operating system drivers on up to application presentation. He has been programming professionally for more than 27 years on platforms ranging from embedded micros up to mainframes and everything in between. He first cut his teeth on programming with a time-share teletype terminal in Fortran IV at the age of 10, and has been hooked ever since. Kevin has programmed in over 30 computer languages and is currently programming in C# and Visual Basic .NET (with a little ILAsm thrown in just for fun). Kevin lives and works near Washington's capital and has served proudly as a United States Marine.

## More Information

For more information about FarPoint Technologies and their FarPoint Spread for Windows Forms solutions, visit them online at <http://www.FarPointSpread.com> or call them at their North American office at 919-460-4551 (or email them at [FPSales@FarPointSpread.com](mailto:FPSales@FarPointSpread.com)). You can reach them at their European office at +44 (0) 1798 812 372 (or email them at [SalesEurope@FarPointSpread.com](mailto:SalesEurope@FarPointSpread.com)).

FarPoint Technologies, Inc. is a leading provider of professional solutions for enterprise-wide development in Microsoft Visual Studio .NET, Microsoft SharePoint, Microsoft BizTalk, and other enterprise platforms. FarPoint set the standard for spreadsheet components with the world's best selling spreadsheet component, Spread, and continues to set high standards in component architecture, offering products across a range of platforms.