

CASE STUDY: ATS Global



“Thanks to PostSharp, our team was able to deliver the required features by writing 16% fewer lines of code than before.”

Michael Hoppe

Software Development Manager
ATS Global

The leading independent solution provider for industrial automation utilizes PostSharp to avoid multithreading issues and saves 16% lines of code.

About ATS Global

www.ats-global.com

ATS Global is a leading independent solution provider for industrial and process automation, quality and information technology. ATS Global has 30 years of experience in the manufacturing systems arena and employs over 400 employees in 23 offices worldwide. The company provides products and services for all three levels of the automation pyramid: control, execution and information. It serves the world's biggest brands including Boeing, Ford, Toyota, Schweppes and Bosch, to name a few. Most software development at ATS Global is done in C++ and C#. In particular, the .NET team is composed of 25 developers located in several countries.



“Thanks to PostSharp, someone who has just a basic understanding of threading is able to write code that is correctly synchronized and thread-safe.”

Michael Hoppe
Software Development
Manager
ATS Global

ATS Global turned to PostSharp to reduce development costs and improve the quality of their software products. The team estimates they delivered required features with 16% fewer lines by utilizing PostSharp.

Preventing random multithreading issues

When the team at ATS Global needed to build a complex shop floor simulation, they faced the challenge of potential multithreading issues and the complexity of writing synchronization code.

A simulation contains many different objects such as machines, materials, or processes such as producing goods, retrieving raw material from storage, etc. All objects are running on separate threads and interact with each other – a situation notoriously prone to random multithreading defects. The team recognized the complexity of the project.

“Having many threads acting as individual objects and synchronizing that is pretty complex, we really thought that we needed a different knowledge level from the developers here in order to implement the synchronization code manually,” said Michael Hoppe, Software Development Manager at ATS Global B.V.

As PostSharp was a proven product that had solved other challenges in

various projects, the team decided to utilize the tool in this project to ensure that their code was thread-safe and synchronized in a predictable and deterministic way – without having to train every team member into a multithreading expert.

The team used several ready-made PostSharp threading models to implement the functionality:

- Reader-Writer Synchronized Threading Model
- Dispatch to UI Thread
- Dispatch to Background

Michael said, “Using PostSharp for patterns like Thread Dispatching and Thread Synchronization allowed developers to do this in a correct way without detailed knowledge of Windows thread handling. It hides aspects of the underlying implementation and let us focus on the high-level implementation.”

PostSharp threading models helped the team identify multithreaded defects deterministically in single-threaded unit tests, as PostSharp validated the code against the threading model. The team avoided multithreading issues and could focus on their core business: shop floor simulation.

“Thanks to PostSharp, someone who has just a basic understanding of threading is able to write code that is correctly synchronized and thread-safe,” said Michael.

Leveraging more of PostSharp patterns

In addition to multithreading, the team identified several other features that would have required developers to write a large amount of repetitive code. They used PostSharp to automate the implementation of `INotifyPropertyChanged`, exception handling, tracing, localization, and undo/redo.

INotifyPropertyChanged

To exchange data between clients and WCF-based servers, the team made use of Data Transfer Objects (DTO). These simple objects stored the data necessary for a specific operation, for instance retrieving a list of all defects in the data store, or persisting a new defect in the data store.

However, the DTOs were also often used as the objects that the user interface is working against. To make the user interface reflect changes made to any of the objects in real time without specifically refreshing, the team had to implement the `INotifyPropertyChanged` interface into

“Using PostSharp for patterns like Thread Dispatching and Thread Synchronization allowed our developers to do this in a correct way without detailed knowledge of Windows thread handling.”

Michael Hoppe
Software Development
Manager
ATS Global

more than 500 classes. Doing this work manually would have required hundreds of hours of repetitive work. Instead, the team chose to use PostSharp’s NotifyPropertyChanged pattern and avoided most of this effort – saving considerable development and maintenance costs.

“Although implementing INotifyPropertyChanged manually would be perfectly possible, it would be a nightmare to maintain in our case. Thanks to PostSharp, it’s fast and clean,” said Tom Bell, Software Development Manager.

Localization

As ATS Global serves customers all over the world, its products are localized in several languages, thus localization typically required a considerable amount of repetitive code.

Using the PostSharp Aspect Framework, the team implemented a custom pattern that consisted of one aspect and one normal attribute.

- An aspect, applied to localizable classes, added new data structures (essentially the ‘localizable string store,’ a nested dictionary).
- A second custom attribute was put on individual localizable properties and was used by the aspect as a marker to determine whether or not to include that property in the ‘localizable string store’ and intercept the property’s getter and setter.

The pattern was used in more than 50 classes and helped the team keep their source code clean and succinct.

Exception Handling

Careful exception handling is of paramount importance for any application deployed to a large number of end users. Exceptions must be properly logged and notified to the user, either in a dialog box or in a system notification icon. Although the exception handling logic itself can be implemented in a central helper class, it is still typically necessary to add try/catch code to hundreds if not thousands of methods.

Instead, the team implemented a custom aspect named HandleException to more than 2,400 methods. This saved the developers from implementing the same logic over and over again wherever exceptions had to be dealt with.

Other Aspects

Additionally, the team leveraged:

- Tracing – The team valued the ability to temporarily add and remove tracing to specific classes for diagnostics and debugging purposes without impact on source code.

- Configuration – The aspect made it easy to expose configuration settings, stored in configuration files, as normal fields and properties.
- Undo/Redo – The team developed a custom aspect to handle their very specific needs, saving hundreds of lines of code.

Results

Thanks to PostSharp, the .NET development team estimates it was able to deliver the required features by writing 16% fewer lines of code than without PostSharp. The code is cleaner and easier to understand, and the applications are easier to maintain thanks to more reusable components. Additionally, the team was able to write thread-safe, machine-verified code and avoided not only most of the random bugs typically found in multithreaded applications, but could do this without training all team members in advanced multithreading.

“We’re very happy since we started using PostSharp and it has proven a truly valuable tool for us. PostSharp makes you rethink your code and you start looking for opportunities to use it more,” said Tom Bell, Software Development Manager at ATS Global B.V.

About PostSharp Technologies

PostSharp is the #1 best-selling pattern-aware extension to C# and VB. It allows developers to eradicate boilerplate by offloading repetitive work from humans to machines. PostSharp contains ready-made implementations of the most common patterns and gives you the tools to build automation for your own patterns.

PostSharp is trusted by over 50,000 developers worldwide and over 10% of all Fortune 500 companies including Microsoft, Intel, Bank of America, Phillips, NetApp, BP, PricewaterhouseCoopers, Volkswagen, Hitachi, Siemens, and Oracle, who rely on PostSharp to reduce their development and maintenance costs.

For more information, please visit <https://www.postsharp.net/>.

SharpCrafters s.r.o.

Namesti 14 rijna, 1307/2
150 00 Prague 5
Czech Republic

US: +1 866 576 5361

CZ: +420 270 007 790

www.postsharp.net

info@postsharp.net