



Building Multi- User Services with AF SDK

Presented by **Ray Verhoeff** and **Brad Hess**

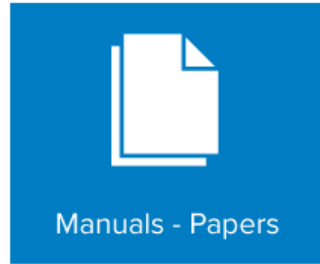




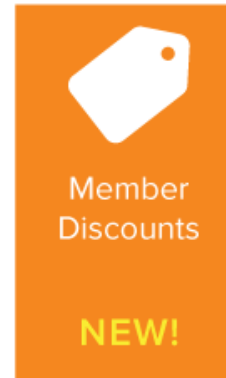
From vCampus to PI Developers Club

Same Vision, Improved Mission

New Model Open to All:



New Model for Paid Users:



Feedback from Partners

- Access to PI System server software is essential
- Forums are great
- The Developer Support team is very dedicated

How can we help?

- Performance metrics
 - OSIssoft's own measurements
 - Tools to help you make your own
- Samples of the Restricted Technologies
- “Samples showing how OSIssoft Engineering thinks the tools should be used”

Advanced AF SDK Software Development

- “Samples showing how OSIsoft Engineering thinks the tools should be used”

Advanced AF SDK Scenarios

- Building web-facing applications on top of AF SDK
- Performing intensive processes in parallel
- Controlling multi-user access in a single application

Pitfalls of Parallel Programming

- In a single-threaded application, AF SDK programming is easy!
- Problems for multi-threaded applications
 - Potential for race conditions
 - Potential for deadlocks

Sample Race Condition

Thread 1

```
AFAttributeContributor.AddAttribute("attribute");  
element.CheckAttributeContributor();
```

InvalidOperationException

*Collection was modified;
enumeration operation may not
execute.*

Thread 2

```
for (int i = 0; i < results.Count; i++)  
{  
    results.AddAttributeContributor(i);  
}
```

Approaches to Handling

- Try-Catch-Retry
- Multiple local AF replicas
- Exclusive Access
 - Monitor
 - ReaderWriterLock
 - ConcurrentExclusiveSchedulerPair

Multiple AF Replicas

- new PISystems(true) forces a new PISystems object with its own cache
- All operations should refer to the context of this PISystems
 - AFObject.FindObject(path, relativeTo)
- Memory concerns
- Performance concerns

Exclusive Access

- Force all operations to acquire some kind of lock in order to access the AF cache.
- You'll need to do this in your own code – there's no AF-provided mechanism.

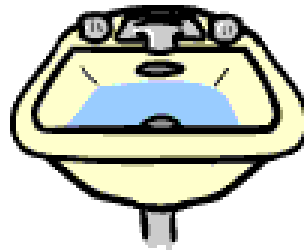
Kinds of Locks

- Monitor
 - Only one consumer at a time
 - Simple semantics in C# with syntactic sugar
 - `lock (target) { }`
- Reader/Writer Lock
 - Multiple concurrent readers
 - Only one writer at a time
 - No readers while there's a writer
 - More complicated to program
- Both are bound to threads in .NET



ConcurrentExclusiveSchedulerPair

- A type of Reader/Writer lock that's implemented as a pair of TPL TaskSchedulers
- Dispatch "readers" to the ConcurrentScheduler, "writers" to the ExclusiveScheduler
- Async-compatible



Working with Multiple Users

- PISystems instances cannot be shared between users
- Connections to PI Data Archive and AF Servers must be made under the correct credentials





DEMO



What Do I Need To Do?

- Determine a strategy for allocation of PISystems instances based on projected needs
- Protect access to allocated PISystems instances to prevent race conditions

What does PI Web API Do?

- PI Web API allocates a PISystems instance per Windows user, and protects access with a ReaderWriterLockSlim
 - We're prototyping an eventual change to ConcurrentExclusiveSchedulerPair
- Just use PI Web API! 😊

Ray Verhoeff

ray@osisoft.com

Product Manager
OSIsoft, LLC

Brad Hess

bhess@osisoft.com

Development Lead
OSIsoft, LLC

Questions

Please wait for the **microphone** before asking your questions

State your **name & company**





THANK YOU

Brought to you by  **OSIsoft.**