Code Contracts

because everyone hates runtime exceptions

Bangkok .NET User Group
2015-10-27
Will Fuqua

C# is good!

static type checking is good!

runtime exceptions are **bad :(**

unexpected system states are **bad :(**

Many bugs are due to mismatched assumptions

```
public int MyLibraryMethod(int[] arr)
{
    // arr will always be length 2
    return arr[1];
}

public int NewFeature()
{
   return MyLibraryMethod(new[] { 4 });
}
```

Runtime Checks are OK...

```
public int MyLibraryMethod(int[] arr)
{
    if (arr.Length < 2)
        throw ArgumentException("arr");
    return arr[1];
}
```

...but it could be better!

Code Contracts helps solve this problem

Document class, method, and interface assumptions

Check assumptions at runtime and **compile-time**

Used extensively inside .NET Framework

Let’s start coding

Code Contracts API

```
void Assert(bool condition);
void Assume(bool condition);
void EndContractBlock();
void Ensures(bool condition);
void EnsuresOnThrow<TException>(bool condition) where TException : Exception;
bool Exists(int fromInclusive, int toExclusive, Predicate<int> predicate);
bool Exists<T>(IEnumerable<T> collection, Predicate<T> predicate);
bool ForAll(int fromInclusive, int toExclusive, Predicate<int> predicate);
bool ForAll<T>(IEnumerable<T> collection, Predicate<T> predicate);
void Invariant(bool condition);
void Requires(bool condition);
void Requires<TException>(bool condition) where TException : Exception;

T Result<T>();
T OldValue<T>(T value);
T ValueAtReturn<T>(out T value);

[ContractInvariantMethod] Attribute
[ContractClassAttribute] Attribute
```

What we’ve learned: it’s not all perfect

Code Contracts slows compilation

Consider running it as part of Continuous Integration
or only on targeted projects

Roslyn broke a lot of things

The Code Contracts team has been fixing things day by day

Thanks! Any questions?

Up next:

F*