What are some instances in which expression trees are useful?

Asked 16 years, 4 months ago Modified 12 years ago Viewed 2k times

11

I completely understand the concept of expression trees, but I am having a hard time trying to find situations in which they are useful. Is there a specific instance in which expression trees can be applied? Or is it only useful as a transport mechanism for code? I feel like I am missing something here. Thanks!

c# expression

Share

Improve this question

Follow

edited Dec 4, 2012 at 8:41

Zerium

17.3k • 32 • 116 • 186

asked Aug 26, 2008 at 10:45

Sean Chambers

8,670 • 7 • 42 • 55

3 Answers

Sorted by: Highest score (default)

Some unit test mocking frameworks make use of expression trees in order to set up strongly typed expectations/verifications. Ie:

7

```
myMock.Verify(m => m.SomeMethod(someObject)); // tells
method
                                                // SomeM
                                                // someO
```

Here, the expression is never actually executed, but the expression itself holds the interesting information. The alternative without expression trees would be

```
myMock.Verify("SomeMethod", someObject) // we've lost
```

Share Improve this answer

Follow

answered Aug 26, 2008 at 10:53

Fredrik Kalseth

14.2k ● 4 ● 26 ● 18

6

> Or is it only useful as a transport mechanism for code?

It's useful as an *execution mechanism* for code. Using the interpreter pattern, expression trees can directly be interpreted. This is useful because it's very easy and fast to implement. Such interpreters are ubiquitous and used even in cases that don't seem to “interpret” anything, e.g. for printing nested structures.

Share Improve this answer

Follow

answered Aug 26, 2008 at 11:03

Konrad Rudolph

545k ● 139 ● 956 ● 1.2k

4

Expression trees are useful when you need to access function logic in order to alter or reapply it in some way.

Linq to SQL is a good example:

```
//a linq to sql statement
var recs (
    from rec in LinqDataContext.Table
    where rec.IntField > 5
    select rec );
```

If we didn't have expression trees this statement would have to return all the records, and then apply the C# where logic to each.

With expression trees that `where rec.IntField > 5` can be parsed into SQL:

```
--SQL statment executed
select *
from [table]
where [table].[IntField] > 5
```

Share Improve this answer

Follow

answered Aug 26, 2008 at 17:57

Keith

155k ● 82 ● 306 ● 446