Eos 0.3 Release Notes

1. Reflective Information

There are a bunch of new interfaces available to extract the reflective information at the join point. In particular, the reflective special variable thisJoinPoint in the advice is now of type Eos.Runtime.IJoinPoint. The signatures at the join points are now available.

1.1 Eos.Runtime.IJoinpoint

The members of this new interface are described below:

This property: Returns the value of "this" at the join point. Returns null in the case of static join points

Target property: Returns the target of the join point. Returns null in cases where there is no target.

ReturnValue property: Returns the return value at the join point if it is a valid return value otherwise null.

Args property: Returns the list of arguments of the join points. Returns an array of length zero if there are no arguments.

Kind property: Returns the string representation of the join point kind.

Signature property: Returns the signature of the join point. Please see the interface Eos.Runtime.Signature.ISignature for more details.

StaticPart property: Returns the static representation at the join point. Please see Eos.Runtime.IStaticPart for more details.

Location property: Returns the Location of the join point. Please see Eos.Runtime.ISourceLocation for more details.

1.2 Eos.Runtime.ISourceLocation

Details of the static location of the join point

EnclosingType property: The System.Type representation of the type that encloses the join point

FileName property: The file name that contains the join point.

Line property: The line number at which the join point occurs.

1.3 Eos.Runtime.IStaticPart

The static part of the reflective information at a join point.

Signature property: The signature of the join point.

Location property: The source location of the join point.

Kind property: The string representation of the join point kind

1.4 Eos.Runtime.Signature.ISignature

Summary description for ISignature.

Name property: Name of the signature.

Modifiers property: Modifiers of this signature

DeclaringType property: Type of the enclosing class

DeclaringTypeName property: Name of the enclosing class

ToString method: Returns a string representation of this signature.

1.5 Eos.Runtime.Signature.ICatchClauseSignature

The signature for the handler join points.

ParameterType property: Returns the type of the exception caught

ParameterName property: Returns the name of the exception caught.

1.6 Eos.Runtime.Signature.IConstructorSignature

The signature of a constructor. An object initialization join point returns this signature.

ParameterTypes property: The list of the types of the parameters at the join point.

ParameterNames property: The list of the names of the parameters at the join point.

1.7 Eos.Runtime.Signature.IFieldSignature

Signature of a field. A field get, set and property get, set join points return this signature.

FieldType property: The System.Type representation of the type of the field.

1.8 Eos.Runtime.Signature.IMethodSignature

Signature of the method. A call and an execution join point return this signature.

ReturnType property: The return type of the method.

2. Declare parents construct:

Similar to AspectJ, aspects in Eos can change the inheritance hierarchy of a system by

1. Either changing the base class of a class (abstract or concrete).
2. Aspects can change the interfaces that a class implements by adding a new interface onto the class.

The syntax of the declare parents construct is as follows:

```
declare parents: type pattern : type list;
```

It is a compile time error for a class to have multiple base classes. A class may implement multiple interfaces.

For example, if an aspect wished to make a particular class clonable, it might introduce the object clone() method into the class, but it should also declare that the class fulfills the System.ICloneable interface. Such an aspect would be:

```
 aspect MakeCloneable {
    declare parents: SomeClass : Runnable;
    introduce in Foo
        {
                public virtual object Clone()
                {
                        return this.MemberwiseClone();
                }
        }
}
```