C# 6.1

Biblical Programming Extensions

Overview

- With the pending release of the C# 6.0 programming language it is time to turn our attention to the features for C# 6.1.
- The C# language team has decided to focus their efforts on adding extensions to the language which support biblical programming practices.
- This presentation outlines a number of compiler and language features which are our early explorations into this area.

New Features

- Bible Version Selection
- The ‘wwjd’ block keyword.
- The ‘pray’ block keyword.
- The ‘miracle’ comparison operator.
- The ‘$$’ tithing arithmetic operator.
- The ‘sabbath’ method modifier.
- New reserved words.
- Gender Aware Code

Bible Version Selection

- Before a programmer can use the biblical programming extensions they must first specify which version of the bible that they are using for syntax parsing.
- The C# compiler (csc.exe) has had the /bible switch added.
- The /bible switch supports a range of arguments, specifically:
  - NIV = New International Version
  - NKJV = New King James Version
  - Others are coming soon.

Bible Version Selection (usage)

csc.exe [regular arguments] /bible:niv

The ‘wwjd’ block keyword.

- The ‘wwjd’ or ‘what would Jesus do’ block keyword allows a developer to provide a list of possible instructions and the compiler will instruct the runtime to seek divine intervention to chose the right option.
- Note that the operator is not guaranteed to select a desirable outcome.

The ‘wwjd’ block keyword (usage)

```
wwjd {
        // Option 1.
} or {
        // Option 2.
} or {
        // Option 3.
}
```

The ‘pray’ block keyword.

- The ‘pray’ block keyword is an assertion construct which allows a developer to specify something that they would like to be true.
- The ‘pray’ block works similarly to the ‘if’ keyword except that a programmer must call the appropriate method depending on the outcome.
- If the ‘pray’ block is successful the programmer must call the ‘PraiseBeToGod()’ method, otherwise the ‘TheLordWorksInMysteriousWays()’ method must be invoked.
- Failure to call either of these methods will result in a runtime exception when biblical extensions are enabled.

The ‘pray’ block keyword (usage)

```
pray (assertion) as myPrayer {
        myPrayer.PraiseBeToGod();
        // Continue processing.
}
else {
        myPrayer.TheLordWorksInMysteriousWays();
        // Compensate for prayer failure.
}
```

The ‘miracle’ exception block keyword

- The ‘miracle’ exception block keyword can be used with either the ‘try’ keyword when it is highly likely that an exception would occur.
- The contents of a miracle block are invoked immediately following the execution of each statement in a try block and gives developers the chance to worship their deity.

The ‘miracle’ exception block keyword (usage)

```
try {
        // Line 1 (successfully executes).
        // Line 2 (successfully executes).
        // Line 3 (throws error).
}
miracle by virginMary {
        virginMary.GiveThanks(); // would be executed three times.
}
catch (Exception ex) {
        // The devil must have done it.
}
```

The ‘$$’ tithing arithmetic operator

- The ‘$$’ or ‘tithing’ operator must be appended to all arithmetic operations in C# when the biblical extensions are enabled.
- The ‘$$’ operator works by deducting 10% from every arithmetic result.
- The help protect against accidental reduction in tithing the operator only works when the resulting value is positive (you’re welcome).

The ‘$$’ tithing arithmetic operator (usage)

```
var a = 100;
var b = 100;
var c = a + c; // Results in compiler error.
var c = a + c $$; // Results in 180;
```

The ‘sabbath’ method modifier

- The ‘sabbath’ method modifier can be added to any method which should not be invoked on a sabbath day.
- If a method marked with the ‘sabbath’ modifier is called on a sabbath day then an EighthCommandmentException will be thrown.
- Recognising that this could be a problem for some developers we’ve also added the ‘football’ modifier which can be wrapped around any method call which suppresses this exception when the need arises.

The ‘sabbath’ method modifier (usage)

```
public sabbath void DeliverBeer(Address address) {
        // Code that ships beer.
}

this.DeliverBeer(); // Would throw an exception on the Sabbath.
football { this.DeliverBeer(); } // Exception suppressed.
```

New Reserved Words

- The following reserved words are also added to the language when the biblical extensions are enabled and cannot be used as variable names, type names or method names.
  - Abort
  - Clone

Gender Aware Code

- In addition, all types must have a gender specified and addition operators cannot be used on two objects of the same gender.
- We realise that this requirement will make some software difficult to write so we've added another command-line switch to disable this feature for those scenario:
  - /ga:off
- … in fact there are a number of such switches planned so that developers can pick and chose their own interpretations of the language syntax.