NET Core Func Variables

Here are some use cases for a Func.

Func as Method Callback

If you want to have a lambda that you pass to a method, like a callback or completion handler, here is an example.

Declare the Func, like this:

```
Func<int, int, Task<int>> sigcallback = async (callbackifr, callbackrds) =>
{
    // The method call, below (Save_Message_toInflightQueue), executes this lambda as a callback, to give us access to the IFR records that were saved.

    // Add the given entry to the listing...
    pairs.Add((callbackifr, callbackrds));
    return 1;
};
```

The above declaration creates a function (as a variable) that accepts two integers as parameter and returns a Task<int>. This means the Func variable works just like a value-Task, when called.

You can then, pass the above Func variable to a method as a callback, like this:

```
// Here, we pass the callback into the method...
var res = await Save_Message(somevariable, sigcallback);
```

The called method has a body like this:

```
public async Task<int> Save_Message(object msg, Func<int, int, Task<int>> signaling_callback)
{
  if(signaling_callback != null)
  {
    // Pass in the current rds and IFR pair...
    var rescallback = await signaling_callback(r.Item2, r.Item1);
  }
```

```
  return (1, (rl.Select(m=>m.Item1).ToArray(), c));
}
```

Revision #1
Created 10 April 2025 03:08:57 by glwhite
Updated 10 April 2025 03:10:12 by glwhite