Logging Integration with Topshelf

By default, Topshelf uses a TraceSource for logging. This is part of the .NET framework, and thus does not introduce any additional dependencies. However, many applications use more advanced logging libraries, such as Logary or NLog. To support this, an extensible logging interface is used by Topshelf.

log4net Integration

To enable logging via log4net, the Topshelf.Log4Net NuGet package is available. Once added to your project, configure Topshelf to use log4net via the configuration:

```
HostFactory.New(x =>
{
    x.UseLog4Net();
});
```

This will change the `HostLogger` to use log4net. There is an overload that allows a configuration file to be specified. If given, the filename will be resolved to the ApplicationBase folder and passed to log4net to configure the log appenders and levels.

Logary integration

To ship logs with Logary, use the Logary.Adapters.Topshelf nuget. Once you’ve added this nuget to your project, you can configure Topshelf to use it via the configuration builder:

```
using (var logary = ...Result)
    HostFactory.New(x =>
    {
        x.UseLogary(logary);
    });
```

This makes it possible to get your logs off your node, so that you avoid running out of disk space and can log to modern log targets, such as ElasticSearch and InfluxDB.

For more information, see the Logary README at https://github.com/logary/logary.