# Design for Trace Event Parsing

This document is meant to capture the design decisions associated with of the set of managed classes focused around manipulating windows event tracing data (currently stored in event tracing log (ETL) files).

# Requirements

The key design requirements

1. Easy to use: One large problem with profile data is that we don’t really understand the best ways to present it usefully. All indications are that domain specific (or even scenario specific) analysis may be needed to useful present the data. Thus it is expected that the library will be used by people who do NOT want to invest time in understanding the API. Ideally the object model for the data is so straightforward, and the ‘common’ operations that users might want to perform are already defined, so that implementing a domain or scenario specific analysis is only minutes of work.
2. Performance and Scaling: Even small data traces measure in the 10MB region, 100MB-1GB traces are very common, 1GB traces are not uncommon, and > 10GB are possible. Thus the data manipulation code must be fast and scalable.

# Design Choices

1. Strongly (and early) typed payloads.
2. Callback and Enumeration model
3. Events as References Classes
4. LINQ support
5. ‘Lazy,Persistant’ objects and a cooresponding file format.