



toothpick



Toothpick – a new logging framework for Smalltalk



What is Toothpick?

- A simple logging framework
- Separates events and concerns
 - Whether to log
 - Where to log
 - What to log



Logger

- Defines where to log
 - Transcript
 - File
 - Socket
 - stdout, stderr
 - NT event log
 - Syslogd, IRC



Logger example

```
| logger1 logger2 |  
logger1 := TranscriptLogger new  
    name: 'My first Logger'.  
logger2 := FileLogger new  
    name: 'My second Logger';  
    fileName: 'MyLog.txt'.
```



LoggingFormatter

- Defines what (how) to log
 - Simple text
 - XML
 - Pattern (like printf())
 - Timestamp
 - syslogd()
 - Log4J ☺



LoggingFormatter example

```
| logger1 logger2 |  
logger1 := TranscriptLogger new  
    format: SimpleLoggingFormatter defaultFormat.  
logger2 := FileLogger new  
    fileName: 'MyLog.txt';  
    format: SimpleLoggingFormatter defaultFormat.
```



LoggingPolicy

- Defines whether to log an event
- Simple
- Composite
 - Multiple event patterns
 - Borrowed from syslogd()



#isInterestingBlock

```
LoggingPolicy>>#defaultIsInterestingBlock
```

```
^ [ :event | event category == self category and: [
    event isLevelHigherThan: self level ]]
```



Categories and Levels

- Category may be any Symbol
- Levels predefined

```
LoggingEvent class>>#allLevels  
  ^#(#all #debug #info #notice #warn #error #crit  
    #alert #emerg #panic #fatal #off #none)
```



LoggingPolicy example

```
| logger1 logger2 |  
logger1 := TranscriptLogger new  
    format: SimpleLoggingFormatter defaultFormat;  
    policy: (LoggingPolicy category: #debug level: #info).  
logger2 := FileLogger new  
    fileName: 'MyLog.txt';  
    format: SimpleLoggingFormatter defaultFormat;  
    policy: (LoggingPolicy category: #debug level: #debug).
```



LoggingMultiplexer

- Central log manager



LoggingMultiplexer example

```
| logger1 logger2 |
logger1 := TranscriptLogger new
    format: SimpleLoggingFormatter defaultFormat;
    policy: (LoggingPolicy category: #debug level: #info).
logger2 := FileLogger new
    fileName: 'MyLog.txt';
    format: SimpleLoggingFormatter defaultFormat;
    policy: (LoggingPolicy category: #debug level: #debug).
LoggingMultiplexer current
    addLogger: logger1;
    addLogger: logger2;
    startAllLoggers
```



LoggingEvent

- An item of interest
- Not related to the #triggerEvent: framework
 - But can be very easily connected
- Information
 - category, level, timestamp, message
 - Can include context and exception info



Sending events

```
LoggingEvent
```

```
  category: #debug
```

```
  level: #warn
```

```
  message: 'using obsolete protocol'.
```

```
LoggingEvent
```

```
  category: #myapp
```

```
  level: #debug
```

```
  message: 'reading ini files'.
```



Configuration

- The framework can be configured from an external file.
- The file can be re-read at run-time to reconfigure the framework.



Where do I get this?

- <http://www.metaprogram.com/Toothpick>