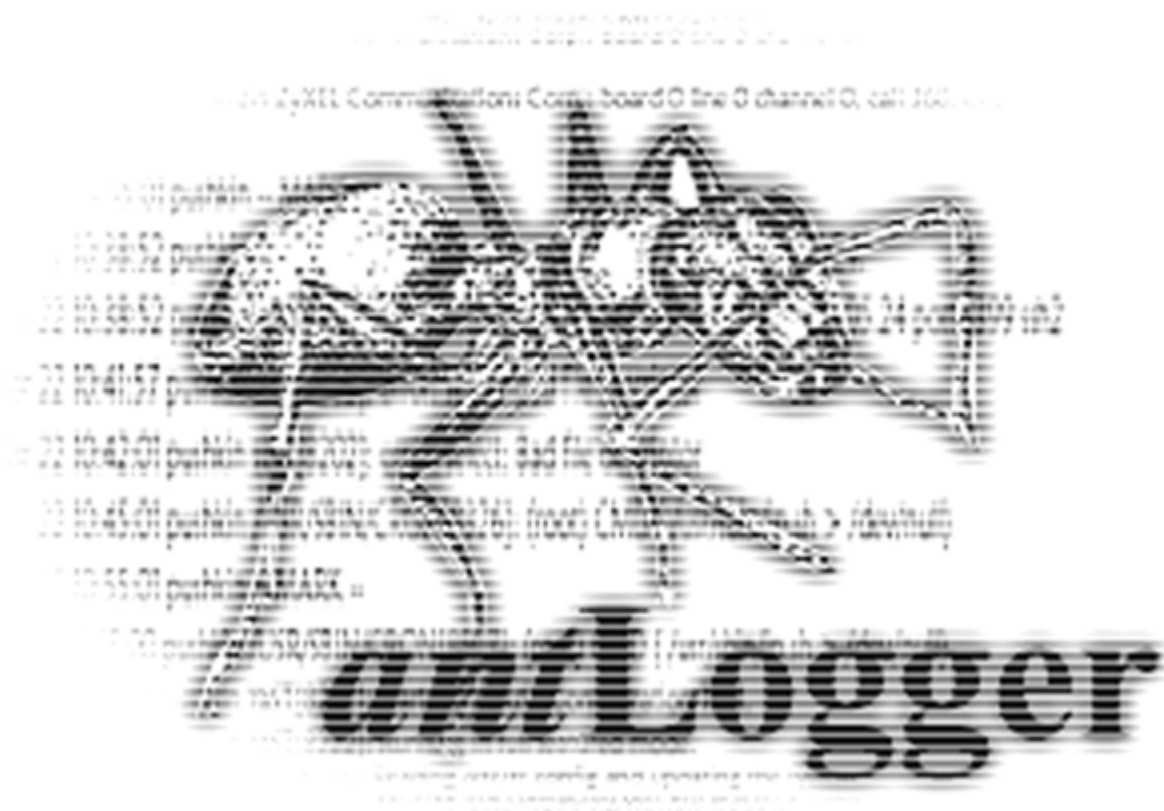




HTA —> HOCHSCHULE FÜR TECHNIK+ARCHITEKTUR LUZERN
Abteilung Informatik->Software Engineering->Projekt AntLogger

Software Engineering Projekt



Software Design Document

Patrick Bründler, Pascal Mengelt, Andy Wyss, Fabian Heusser
Dozent: Jörg Hofstetter

History

Nr	Datum	Name	Beschreibung
1	01.04.2002	Wys	Erstellung
2	22.04.2002	Wys	Änderungen am LogServer Paket
3	07.05.2002	Wys	Korrekturen

Inhaltsverzeichnis

1.	Einleitung.....	3
1.1	Absicht.....	3
1.2	Umfang	3
1.3	Definitionen, Akronyme and Abkürzungen	3
2.	Referenzen.....	3
3.	Zerlegungsbeschreibung	4
3.1	Modulzerlegung	4
3.1.1	antLogger Paket	5
3.1.2	logClient Paket	5
3.1.3	logServer Paket.....	6
3.1.4	testApps Paket	7
4.	Abhängigkeitsbeschreibung.....	8
5.	Schnittstellenbeschreibung.....	8

1. Einleitung

1.1 Absicht

In diesem Dokument werden die Pakete und Klassen des Projektes antLogger beschrieben.

1.2 Umfang

Hier wird der antLogger in komplettem Umfang gezeigt. Dazu gehören Schnittstellen- und Klassenbeschreibungen, sowie ihre Zusammenfassung zu Paketen.

1.3 Definitionen, Akronyme and Abkürzungen

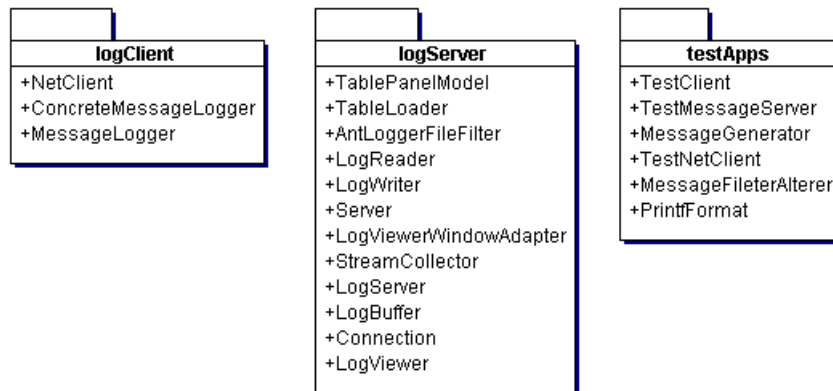
antLogger: Das Hauptpaket, welches alle Elemente für einen voll funktionsfähigen Message-Logger bereitstellt. (siehe auch SRS)

2. Referenzen

Software Engineering: an Object-oriented perspective, by E. J. Braude, Wiley; 2001.

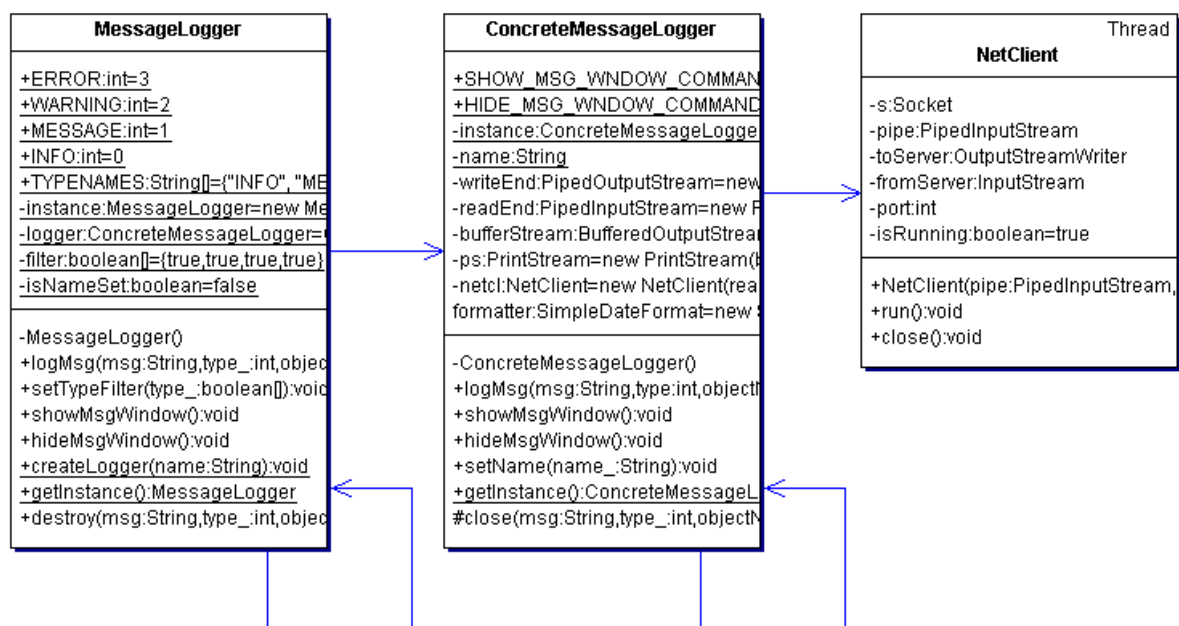
3.1.1 antLogger Paket

Die komplette Applikation antLogger, welche aus einem Server- und einem Client-Package besteht.



3.1.2 logClient Paket

Die Klasse MessageLogger wird als einzige von den Clients aufgerufen.



3.1.4 testApps Paket

Dieses Paket enthält verschiedene Testprogramme.

TestNetClient
-stdin:boolean=true
+TestNetClient() <u>+main(args:String[]):void</u>

TestClient
<u>-level:long</u> <u>running:boolean=true</u>
+TestClient() <u>+main(args:String[]):void</u>

PrintfFormat
-vFmt:Vector=new Vector() -cPos:int=0 -dfs:DecimalFormatSymbols=null
+PrintfFormat(fmtArg:String) +PrintfFormat(locale:Locale,fmtArg:String) -nonControl(s:String,start:int):String +sprintf(o:Object[]):String +sprintf():String +sprintf(x:int):String +sprintf(x:long):String +sprintf(x:double):String +sprintf(x:String):String +sprintf(x:Object):String
-ConversionSpecification

TestMessageServer
-port:int=1234 -input:InputStreamReader
+TestMessageServer() <u>+main(args:String[]):void</u>

MessageGenerator	Thread
threadCount:int=0 msgCount:int=0 <u>+EASY:long=2000</u> <u>+MEDIUM:long=1000</u> <u>+HARD:long=500</u> <u>+INSANE:long=100</u> thisThreadNumber:int type:int level:long m:MessageLogger	
+MessageGenerator(level:long,type:int) +run():void	

MessageFileterAlterer	Thread
threadCount:int=0 -intervall:long=2000 -random:boolean=false -statepointer:int=1 -state:boolean[]={ true, true, true, true } m:MessageLogger	
+MessageFileterAlterer(intervall:long) +run():void +byteToString(statepointer:int):String	

4. Abhängigkeitsbeschreibung

Das LogClient Paket funktioniert nicht ohne einen instanziierten LogServer aus dem LogServer Paket. Der LogServer muss dazu vor dem LogClient gestartet werden.

5. Schnittstellenbeschreibung

Als Schnittstelle für den Client dienen ausschliesslich die public Methoden der Klasse MessageLogger aus dem Paket LogClient. (siehe 3.1.2)