

WHITEPAPER

Date: 2008-07-11

LOG FILE ANALYZER TOOL

Introduction.....	2
Functions	2
System overview.....	2
Subsystems.....	3
Subsystem – Reader.....	3
Brief description of program	3
Function of program.....	3
Functional structure.....	4
Application	4
Program structure.....	4
Program elements.....	4
Description of program sequence	5
Subsystem - Analyzer	5
Brief description of program	5
Function of program.....	5
Data organization / Prerequisites	7
Functional structure.....	7
Program sequence	8
Configuration	9
Subsystem - WebFrontend	9
Brief description of program	9
Function of program.....	9
Description of program sequence	9
Configuration	10
Subsystem - Database	10
Program structure.....	11

Introduction

This tool was developed in cooperation with our partner Inventek to provide dynamic access to log files for various applications and to enable searching in these files according to defined rules, and consequently issue e.g. alarms in monitoring tools.

Functions

One/Number of periodically recorded log file will be read, evaluated/analyzed according to defined rules and the results will be displayed in Web-Frontend. Based on the result it is possible to execute additional actions, such as send an email, write event logs (event viewer) or generate Error-Log file.

System overview

The entire "LogAnalyzer" system consists of number of subsystems.

- Reader
- Analyzer
- Web-Frontend
- Database

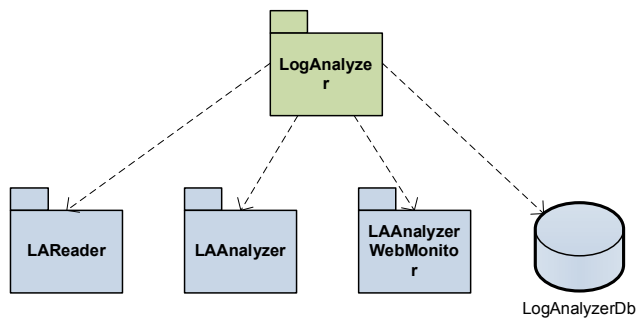


Fig. 1 Total system

Subsystems

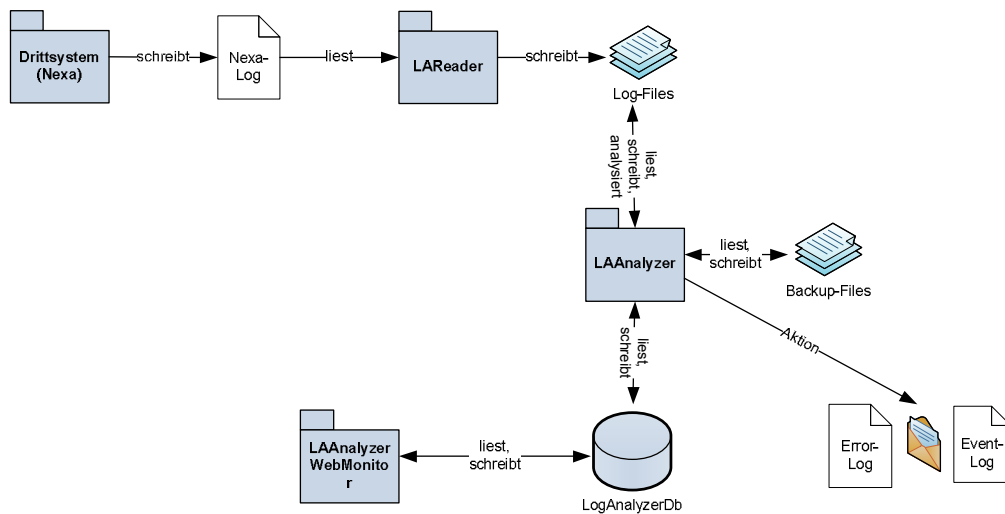


Fig. 2 Subsystems

Subsystem – Reader

Brief description of program

Reader synchronizes log file with local log files created by Reader.

Function of program

Reader reads records from log and writes them in one or more local log file(s). Size of log file may be predefined in configuration file.

Only newly attached record (at the end of file) will be read and synchronized within log.

Reader is designed as console program, as well as service application.

Functional structure

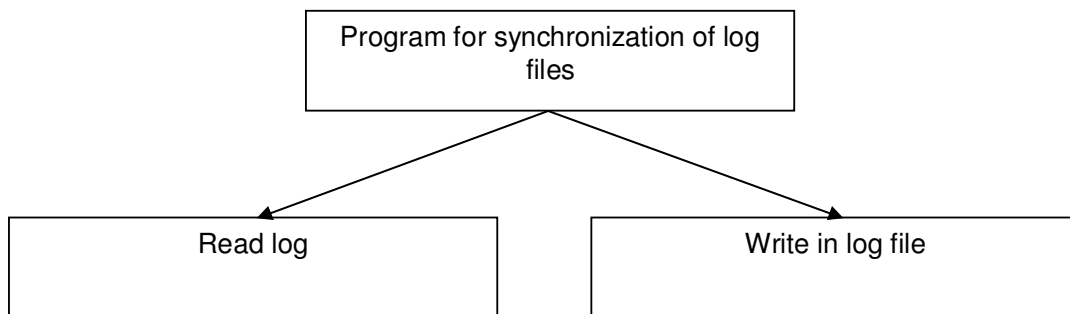


Fig. 3 LARReader functional structure

Application

Reader must be installed as service and preset for automatic starting. Reader works continuously and uses processor capacity only in case of writing into file.

Program structure

Program elements

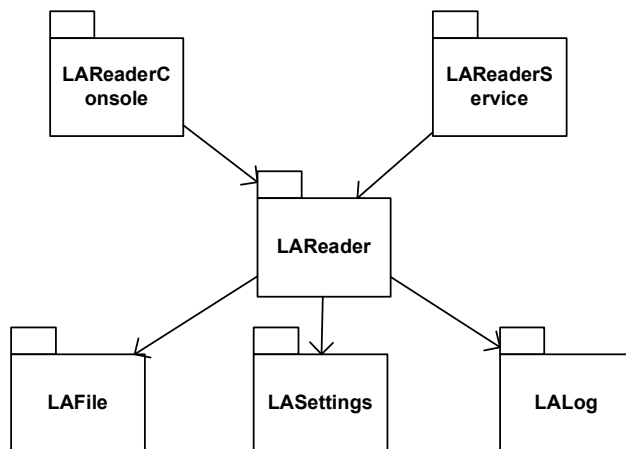


Fig. 4 Program elements of LARReader

Description of program sequence

1. Process of synchronization starts in own Thread.
2. Read old Offset value from *.ini – file.
3. It will be checked, if log exists. If not, system will wait.
4. Location in directory and file name may be modified in configuration file.
5. If the log exists, it will be checked, if new data for synchronization are available.
6. If no data to write are available, system will wait.
7. If the data to write are available, synchronization will be realized.
8. File is opened - not exclusively.
9. File is read.
10. Available names of local log files are loaded.
11. Size of file will be analyzed, eventually new log file created and number assigned.
12. Open log file exclusively.
13. Write in log file.
14. Close log file.
15. Save up-to-date file offset in *.ini - file. Necessary to define new data added to log.
16. Items 8 – 15 should be repeated until the end of file in log is reached.
17. Log will be locked
18. Items 3 – 17 should be repeated for the whole time period, when Thread is active.

Subsystem - Analyzer

Brief description of program

Analyzer evaluates the log records according to defined rules.

Function of program

Analyzer analyses log file(s) in relation to every rule to find appropriate records. Such records will be then saved in database. Rules for searching will be saved in database, as well. For each rule it is possible to define additional actions (e-mail, event log, error log, etc.), which will be executed after finishing evaluation. When the analysis was finished, analyzed log file(s) will be transferred to Backup directory.

Analyzer is designed as console program, as well as service application.

Fig. 5 Function of LAAalyzer

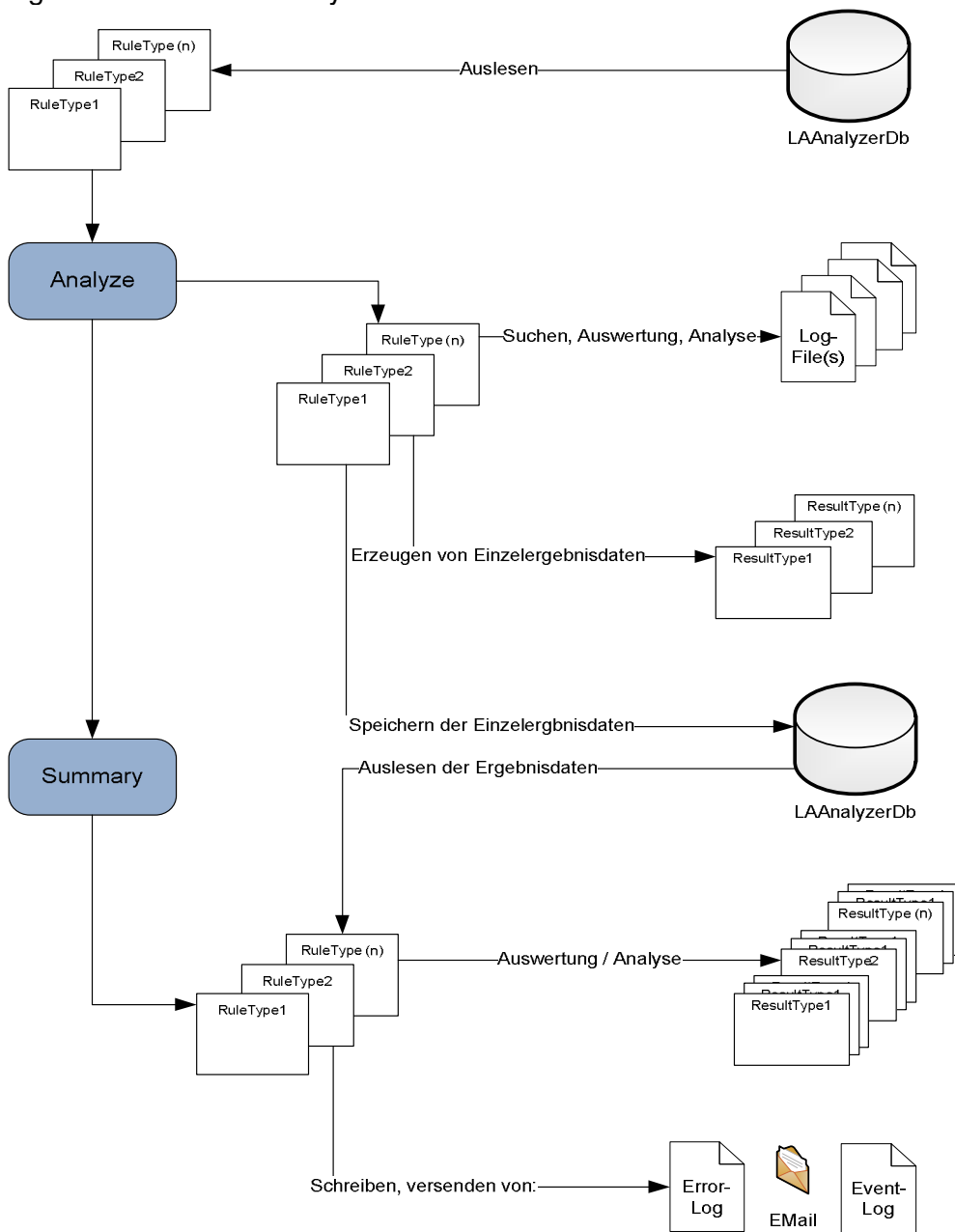


Fig. 6 Function of LAAalyzer (detailed)

Data organization / Prerequisites

Analyzer opens local files exclusively. When the analysis of log records was finished, log files will be transferred to Backup directory. This backup directory may be deleted only by user intervention.

Functional structure

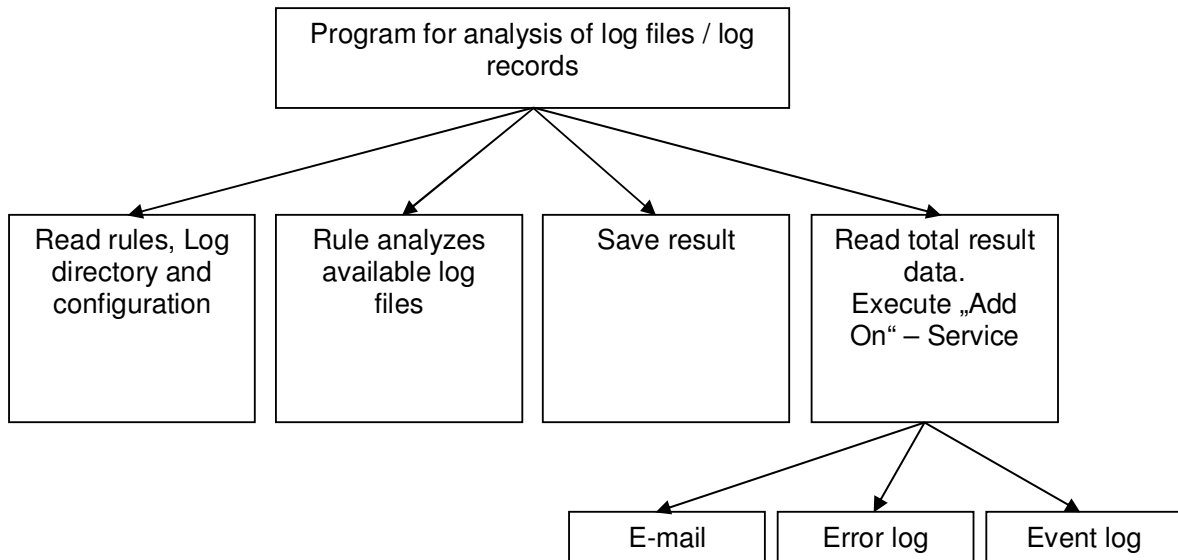


Fig. 7 LAAnalyzer functional structure

Program sequence

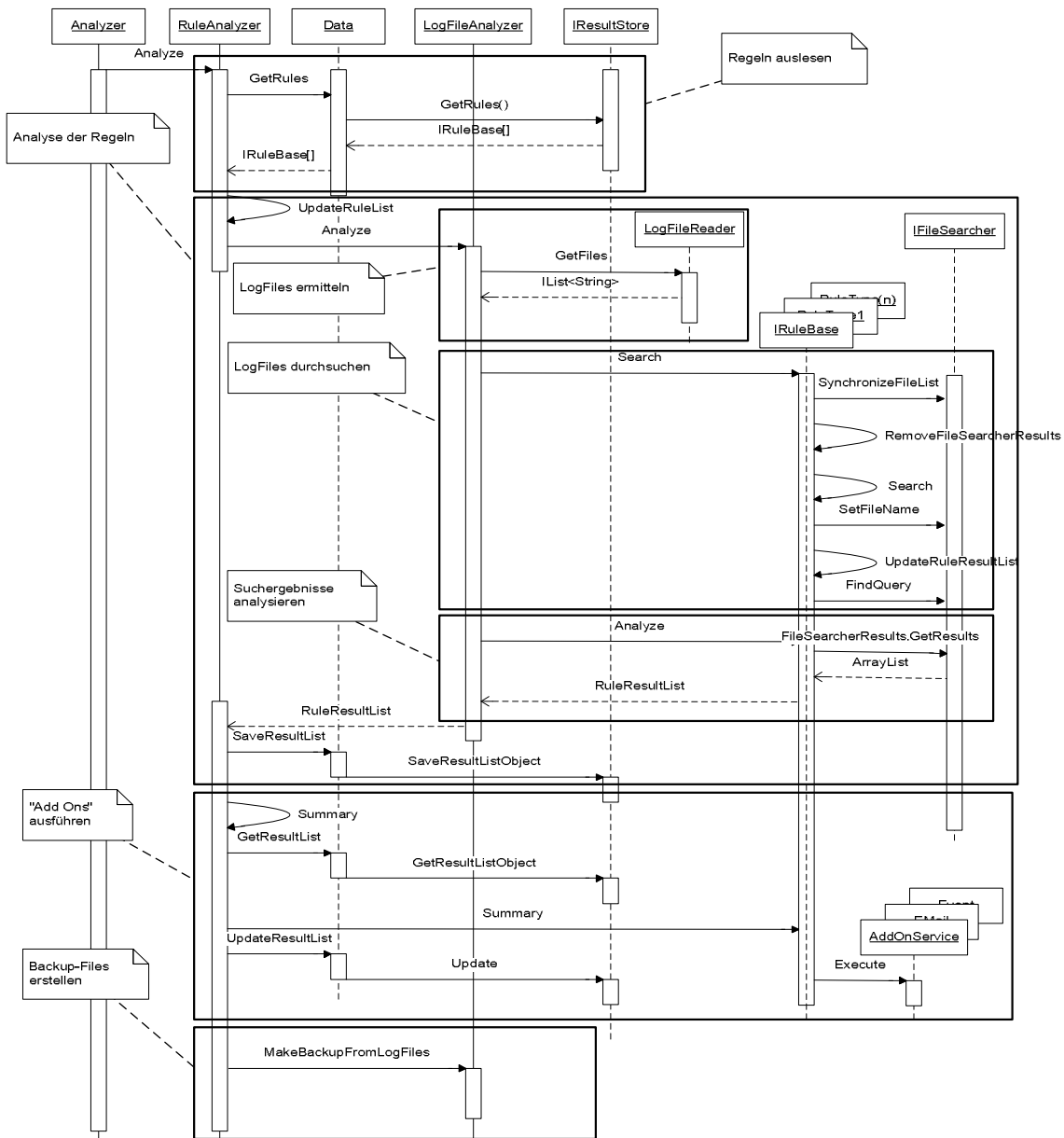


Diagram 8 program sequence of LAAnalyzer

Configuration

- LAAnalyzerSettings.xml
- LAEventLogSettings.xml
- LAHeartBeaterSettings.xml
- LAMailSettings.xml
- LAResultStoreDBSettings.xml
- LAResultStoreSettings.xml
- LARulesSettings.xml
- LASearcherSettings.xml

See Chapter „System - Configuration“.

Subsystem - WebFrontend

Brief description of program

Web-Frontend to display particular result data and to configure users and rules.

Function of program

Web-based frontend displays searching results i.e. found relevant log records to be displayed. Frontend provides also the option highlight, eventually delete records, which are not necessary any more, or were processed successfully.

User must be signed in to confirm displayed events (logic "Delete"). For this reason simple user administration must be implemented.

Results of evaluation will be displayed in form of list.

Description of program sequence

Principle:

1. Actions executed by user, as well as displayed single result data, inserting rules, etc. are processed through central classification of data.
2. Standard "ASPNETDB" database will be used for user administration. Each access is administered through the class "UserData".

Configuration

Following settings will be made using Web- Config:

Element	Description
PathToAppData	Path to configuration files
UpdateTime	(1s => 1000), Updating period (partial update)
ScriptManagerAsyncPostBackTimeout	Asyn. Post Back Timeout of Script Manager

Subsystem - Database

Database saves the result data (relevant log records) and rules.
For user administration is assigned specific database.

