# SVN-ALFEventEmitter

## Sending ALF events from a Subversion hook


### Abstract

It is common to reference ticket numbers within a SVN commit message to indicate the root cause of change in the repository. This software can be used as a Subversion hook to parse the commit message and send the ticket numbers along with other commit information as an xml event to a Serena ALF Event Manager.



Brian Rosenberger

brosenberger@serena.com

# CONTENTS



This document has been created January 2nd 2014.

Last modified December, 02nd 2014.

| | | |
|---|---|---|
| 22.01.2014 | BRO | Added configuration properties “svnlook” and “isWithMessageUpdate” to the emitter.properties |
| 02.12.2014 | BRO | Added encryption configuration properties and chapter 3.1.3 Encrypting passwords |

# 1 LICENSE

```
/*
 * Copyright 2014 Brian Rosenberger (Brutex Network)
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *     http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
*/
```

# 2 INSTALLATION

## 2.1 PREREQUISITES

This software requires a Java Virtual Machine (JVM) for Java 6 or later.

Verified JVMs are:

Any version before 02.12.2014

- java version "1.6.0_27"
  OpenJDK Runtime Environment (IcedTea6 1.12.6) (6b27-1.12.6-1ubuntu0.12.04.4),
  OpenJDK 64-Bit Server VM (build 20.0-b12, mixed mode)

- java version "1.6.0_14"
  Java(TM) SE Runtime Environment (build 1.6.0_14-b08),
  Java HotSpot(TM) 64-Bit Server VM (build 14.0-b16, mixed mode)

- java version "1.7.0_25"
  Java(TM) SE Runtime Environment (build 1.7.0_25-b17),
  Java HotSpot(TM) 64-Bit Server VM (build 23.25-b01, mixed mode)

Versions release after 01.12.2014

- java version "1.7.0_25"
  Java(TM) SE Runtime Environment (build 1.7.0_25-b17)
  Java HotSpot(TM) 64-Bit Server VM (build 23.25-b01, mixed mode)

Subversion:

- svn, Version 1.6.17 (r1128011), Ubuntu 12.04.3 LTS, Kernel 3.2.0-53-generic

## 2.2 WINDOWS

Tbd.

## 2.3 LINUX

Extract the SVN-ALFEventEmitter using

tar –xvf SVN-ALFEventEmitter.tar.gz

The target directory must be readable by all local users who are doing commits against the local Subversion repository. If a Subversion is exposed via TCP/IP (i.e. using Apache HTTPD), then the user who is running the commit process needs to have read privileges to the install directory.

### 2.3.1 Creating the pre-commit hook

The SVN hooks are located under the repository path in the “hooks” folder. Make sure this folder contains a file named “pre-commit” that is executable (“chmod a+x pre-commit”). If that file is not present already, it can either be created from a template (“cp pre-commit.tmpl pre-commit”) or blank using “touch pre-commit”. The content of that file needs to contain a call to the SVN-ALFEventEmitter.

Example content:

```
#!/bin/sh


# PRE-COMMIT HOOK
#
# The pre-commit hook is invoked before a Subversion txn is
# committed.  Subversion runs this hook by invoking a program
# (script, executable, binary, etc.) named 'pre-commit', with the
# following ordered arguments:
#
#   [1] REPOS-PATH   (the path to this repository)
#   [2] TXN-NAME     (the name of the txn about to be committed)
#   [STDIN] LOCK-TOKENS ** the lock tokens are passed via STDIN.
#
JAVA=/usr/lib/jvm/java-6-openjdk-amd64/jre/bin/java
REPOS="$1"
TXN="$2"
$JAVA -jar /path/SVN-ALFEmitter.jar -repos $REPOS -txn $TXN 1>&2
```

You need to adjust the highlighted areas in the shell script. Note that the stdout is piped into stderr (1>&2). This will throw all debug and error messages back to the SVN client in case of any error and the commit will fail.
An “exit 0” command at the very end of the “pre-commit” hook will force to commit to succeed, even when errors did occur.

### 2.3.2 Configuring java

In order to execute, a JVM must be available on the system path. This can be verified by typing ‘java –version’

into console. If an appropriate java version is not on the path, the full system path to the java executable must be given in the “pre-commit” hook file for the “JAVA” variable.

# 3 CONFIGURATION

### 3.1.1 Configuring emitter.properties

By default, the SVN-ALFEventEmitter will use a configuration file “emitter.properties” located in its installation directory. If a single installation is used with multiple repositories, the location of the properties file can be overridden using the “--config <file>” command line parameter.

Configuration example:

```
# Properties for SVN-ALFEmitter
#
# Location of the svnlook utility
svnlook=/usr/bin/svnlook

# RegEx expressions to identify issue ids in the commit message. You can specify multiple
# patterns by repeating the definition following line.
# IDs beginnen mit '#' gefolgt von 6 Ziffern
# IDs beginnen mit DOC, BUG oder ENH direkt gefolgt von 6 Ziffern
issuepattern=#[0-9]{6}
issuepattern=(DOC|BUG|ENH)[0-9]{6}

# A Template XML File for the ALFEvent. This must be a valid XML file
eventtemplate= /path/event-template.xml
eventnamespace=http://www.eclipse.org/alf/schema/EventBase/1

# ALF Eventmanager URL and authentication
eventmanager = http://hostname:8085/eventmanager/services/ALFEventManagerDocLit
eventmanager.user = admin
eventmanager.password =

# Debug options
```

*# force fail is not active when empty*

*forcefail=true*

Other configuration parameters:

| Name | Default | Description |
|---|---|---|
| *eventmanager* | | The URL to the SBM event manager. This is a mandatory property except sending soap messages is disabled using the isSoapEnabled property |
| *eventmanager.password* | | Password for the eventmanager.user |
| *eventmanager.encrypted* | false | Whether or not the "*eventmanager.password*" is encrypted |
| *eventmanager.user* | | The SBM user to be injected into the ALFSecurity context |
| *eventnamespace* | yes | The event xml namespace, Defaults to '*http://www.eclipse.org/alf/schema/EventBase/1*' |
| *eventtemplate* | | An XML file to be uses as template for the SOAP request |
| *isDropResponse* | true | Should the emitter receive the soap response from the event manager and log it to STDOUT (debug mode only). A value of 'true' will drop the response. |
| *isSoapEnabled* | true | Send the event to the event manager. Xmlprocessing must be switched on for this to work. |
| *issuepattern* | | A regular expression to match issue ids within the commit message. You can specify this parameter multiple times to apply multiple regular expressions. |
| *isWithVerification* | false | Enables verification if referenced issues do exists in SBM |
| *isWithMessageUpdate* | false | Enables to modification of the original svn commit message<br>Note: this option cannot be used in conjunction with a pre-commit hook |
| *isXmlProcessingEnabled* | true | Process XML soap template and generate soap message |
| *marker.addedfiles* | @@addedfiles@@ | XML comment marker for the new files list |

| | | |
|---|---|---|
| *marker.author* | @@author@@ | XML comment marker for the commit author |
| *marker.changedfiles* | @@changedfiles@@ | XML comment marker for the changed files list |
| *marker.deletedfiles* | @@deletedfiles@@ | XML comment marker for the deleted files list |
| *marker.fileselementname* | file | The name of the XML element that will wrap each file in the files list. |
| *marker.internalissues* | @@internalissues | XML comment marker for the internal issues list. Only available after a verification phase |
| *marker.issueselementname* | issue | The name of the XML element that will wrap each issue in the issues list. |
| *marker.issues* | @@issues@@ | XML comment marker for the issues list |
| *marker.logmessage* | @@logmessage@@ | XML comment marker for the commit message |
| *marker.revision* | @@revision@@ | XML comment marker for the revision. This marker is only available in a post-commit hook |
| *query* | | Additional where clause |
| *querytable* | | The database name of the primary table in SBM to perform the query against |
| *removeissuesfromcommit* | false | Should we remove the matched issue identifiers from the commit message |
| *sbmappservices72* | http://localhost/gsoap/gsoap_ssl.dll?sbmappservices72 | The sbmappservices72 SOAP endpoint |
| *sbmpassword* | | Password for user specified by "sbmuser" |
| *sbmencrypted* | false | Whether or not the "sbmpassword" is encrypted |
| *sbmuser* | | A user to query issues in SBM |
| *svnadmin* | | Location of the svnadmin utility. |
| *svnlook* | | Location of the svnlook utility. (Linux svnlook, Windows svnlook.exe). Note that you need to properly escape "\" and spaces. Example: svnlook = C:\\Program\\Files\ (x86)\\Subversion\\bin\\svnlook.exe |
| *trace* | false | Log raw XML messages on the sbmappservices72 communication layer. Attentions, reveals passwords. |
| *env.LANG* | de_DE.UTF-8 | Locale that is pushed into a system variable "LANG" before the svnlook command is executed |

| | | |
|---|---|---|
| *env.encoding* | UTF-8 | An encoding supported by java to translate STDOUT and STDERR console streams into strings. This setting should match your env.LANG property. |

### 3.1.2 The ALFEvent XML Template

The SVN-ALFEventEmitter reads an XML file that is used as a basis to construct the ALFEvent. This file should be a valid XML structure containing XML Comment markers. These markers are replaced by information from the Subversion commit attempt.

Available markers:

| | |
|---|---|
| @@timestamp@@ | Inserts current date and time as ISO8601 formatted string. |
| @@logmessage@@ | Inserts the commit message wrapped into a CDATA section. |
| @@author@@ | Inserts the commit user as string. |
| @@addedfiles@@ | |
| @@deletedfiles@@ | |
| @@updatedfiles@@ | All three are lists of files and directories that will change on commit. The lists are wrapped into <file> elements. |
| @@issues@@ | A list of issues identified in the commit message, wrapped into <issue> elements. |

To use a marker, wrap the marker into an XML comment and place it where the replaced content should appear, example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns="http://www.eclipse.org/alf/schema/EventBase/1">
  <soapenv:Header/>
  <soapenv:Body>
    <ns:ALFEventNoticeDoc version="1.0">
      <ns:Base>
        <ns:EventId></ns:EventId>
        <ns:Timestamp><!--@@timestamp@@--></ns:Timestamp>
        <ns:EventType>commit</ns:EventType>
        <ns:ObjectType>CommitInfo</ns:ObjectType>
        <ns:ObjectId></ns:ObjectId>
        <ns:Source>
          <ns:Product>SVN-Hook</ns:Product>
          <ns:ProductVersion>1.0</ns:ProductVersion>
          <ns:ProductInstance>Default</ns:ProductInstance>
        </ns:Source>
      <ns:User/><!—SVN-ALFEventEmitter will automatically inject ALFSecurity Token-->
      </ns:Base>
```

```
      <ns:Extension>
        <ns:issues><!--@@issues@@--></ns:issues>
       <ns:internalissues><!--@@internalissues@@--></ns:internalissues>
        <ns:submitter><!--@@author@@--></ns:submitter>
        <ns:info><!--@@logmessage@@--></ns:info>
        <ns:CommitFiles><!--@@addedfiles@@--><!--@@changedfiles@@--></ns:CommitFiles>
      </ns:Extension>

    </ns:ALFEventNoticeDoc>
  </soapenv:Body>
</soapenv:Envelope>
```

It is possible to override marker keys and the way lists are wrapped into XML elements in the configuration file using the “marker.*” parameters.

The corresponding example output:

*svn commit ./* -m "Here is the commit message for the issues enh012345 hello #000001 DOC453423 somewhere in the text. Note that patterns are case sensitive."*
Sende blabber.txt
Sende blubber.txt
Hinzufügen new.txt
Übertrage Daten ...

```
<?xml version='1.0' encoding='UTF-8'?><soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns="http://www.eclipse.org/alf/schema/EventBase/1">
  <soapenv:Header/>
  <soapenv:Body>
    <ns:ALFEventNoticeDoc version="1.0">
      <ns:Base>
        <ns:EventId>157b1e14-ea42-4778-9f1f-03e859c73909</ns:EventId>
        <ns:Timestamp><!--@@timestamp@@-->2013-12-30T15:48:29.119+01:00</ns:Timestamp>
        <ns:EventType>commit</ns:EventType>
        <ns:ObjectType>CommitInfo</ns:ObjectType>
        <ns:ObjectId>18-42</ns:ObjectId>
        <ns:Source>
          <ns:Product>SVN-Hook</ns:Product>
          <ns:ProductVersion>1.0</ns:ProductVersion>
          <ns:ProductInstance>Default</ns:ProductInstance>
        </ns:Source>
      <ns:User><!--Generated by SVN-ALFEmitter--
><ns:ALFSecurity><ns:UsernameToken><ns:Username>admin</ns:Username><ns:Password><![CDATA[]]><
/ns:Password></ns:UsernameToken></ns:ALFSecurity></ns:User>
```

```
      </ns:Base>

      <ns:Extension>
        <ns:issues><!--@@issues@@-->
<ns:issue>DOC453423</ns:issue><ns:issue>#000001</ns:issue></ns:issues>
<ns:internalissues><!--@@internalissues@@-->
<ns:issue>1000:33</ns:issue><ns:issue>1000:45</ns:issue></ns:internalissues>
        <ns:submitter><!--@@author@@-->root</ns:submitter>
        <ns:info><!--@@logmessage@@--><![CDATA[Here is the commit message for the issues enh012345
hello #000001 DOC453423 somewhere in the text. Note that patterns are case sensitive.]]></ns:info>
        <ns:CommitFiles><!--@@addedfiles@@--><ns:file>new.txt</ns:file><!--@@changedfiles@@--
><ns:file>blubber.txt</ns:file><ns:file>blabber.txt</ns:file></ns:CommitFiles>
      </ns:Extension>

    </ns:ALFEventNoticeDoc>
  </soapenv:Body>
</soapenv:Envelope>
```

### 3.1.3 Encrypting passwords

It is possible to encrypt SBM and eventmanager passwords in the configuration file. The encryption is done via a static key. In order to get you encrypted password value, use the following command:

```
java  -cp <path>\SVN-ALFEmitter.jar net.brutex.emitter.util.PasswordEncrypter <yourpassword>
```

# 4 TROUBLESHOOTING

## 4.1 INVOCATION ON THE COMMAND LINE

The SVN-ALFEventEmitter can be called on a command line directly:

```
usage: java -jar SVN-ALFEventEmitter [-conf <config_file>] –repos
     <repository> [-rev <revision>] [-txn <transactionid>]

SVN-ALFEventEmitter 0.1, a SVN hook implemented in Java to emit Eclipse
ALFEvents on commit.
```

| | |
|---|---|
| -conf,--config <config_file> | The configuration file to use.<br>Defaults to 'emitter.properties'. |
| -repos,--repository <repository> | Path or URL to the SVN repository. |
| -rev <revision> | A revision to examine. You cannot<br>combine revision with -txn. |
| -txn,--transaction <transactionid> | The SVN transaction id to examine (TXN).<br>You cannot combine txn with –rev option. |

*When a txn is given, the repository path must be a local path.*

When calling the SVN-ALFEventEmitter directly, use the –rev option with any existing revision instead of the transaction id. An XML event will be generated and send according to the settings in the configuration file.

## 4.2 FORCE DEBUG

The configuration file has a parameter “forcefail”, any non-empty value will force to Subversion hook to always exit with a return code “1” and thus block the commit. Any other operation is performed, the ALF event is sent. This is useful to test the hook, but without changing the repository head revision.

The configuration parameter “isSoapEnabled=false” can be used to disable the ALF event to be sent to the event manager. The event XML is printed to the stdout as it would have been sent.

## 4.3 CONNECTION TO A REMOTE REPOSITORY DOES NOT WORK

Remote repositories (i.e. via http) are only supported with the “-rev” option. It is also worth to check that your svn client/ svnlook utility supports the desired bindings.

- svnlook --version