ICT PROJECT 258109
Monitoring Control for Remote Software Maintenance

<table>
<thead>
<tr>
<th>Project number:</th>
<th>258109</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Acronym:</td>
<td>FastFix</td>
</tr>
<tr>
<td>Project Title:</td>
<td>Monitoring Control for Remote Software Maintenance</td>
</tr>
<tr>
<td>Call identifier:</td>
<td>FP7-ICT-2009-5</td>
</tr>
<tr>
<td>Instrument:</td>
<td>STREP</td>
</tr>
<tr>
<td>Thematic Priority:</td>
<td>Internet of Services, Software and Virtualization</td>
</tr>
<tr>
<td>Start date of the project:</td>
<td>June 1st, 2010</td>
</tr>
<tr>
<td>Duration:</td>
<td>33 months</td>
</tr>
</tbody>
</table>

D5.3: 1st prototype of the execution recorder/replayer tool

Lead contractor: INESC ID
Author(s): João Garcia, Paolo Romano, Luís Rodrigues, Nuno Coração, Nuno Machado, João Matos
Submission date: May 2011
Dissemination level: Public

Project co-founded by the European Commission under the Seventh Framework Programme
© S2, TUM, LERO, INESC ID, TXT, PRODEVELOP
Abstract: This document describes the design and usage of the execution recorder/replayer components of the FastFix project’s platform. These components are used for providing enhanced error reporting by recording application execution and therefore providing a reliable fault replication mechanism to help maintenance teams using the FastFix platform.
Contents

1 Introduction ............................................................................................................................. 4
2 Overview ................................................................................................................................... 5
  2.1 Code Metrics .................................................................................................................... 5
3 Execution Recorder Tool .................................................................................................. 7
4 Execution Replayer Tool .................................................................................................... 8
5 Running FastFix Record-Replay ...................................................................................... 10
  5.1 Running FastFix Record-Replay as an End-User .......................................................... 10
    A. On the FastFix Server Machine: .................................................................................. 10
    B. On the “Client Machine”: ......................................................................................... 10
  5.2 Running FastFix Record-Replay as a Developer ......................................................... 13
    5.2.1 Infrastructure for FastFix Development ............................................................... 13
    5.2.2 Build and Run FastFix .......................................................................................... 14
      5.2.2.1 Build ............................................................................................................. 14
      5.2.2.2 Run existing bundles ..................................................................................... 15
1 Introduction

A major motivation for the FastFix project is the realization that software maintenance teams, when debugging deployed applications, would often be aided by additional information besides the point of failure and core dump information that is traditionally available in error reports.

The recorder/replayer tool of FastFix plays a central role in supporting FastFix applications. The recorder tool, as its name implies, records the execution of FastFix-enabled applications in order to allow the application’s maintenance team to observe the sequence of user actions that led to the application’s error.

Once an error is detected by the application runtime or the user requests that an error report be sent, the FastFix system on the client device sends an error report to the maintenance team’s FastFix server containing a recorded trace of the application execution.

With the FastFix error report, the maintenance team can replay the execution and observe how the application error occurred and therefore more easily identify the error’s cause.

This document is a companion to the source code of the recorder/replayer tool in the FastFix platform. The following sections detail the design and usage of these components.
2 Overview

The recorder/replayer tool of FastFix is divided in client and server components. On client devices, FastFix records the execution of applications that are configured to use FastFix and enables their users to send detailed error reports to the maintenance team. The current version of the recorder tool monitors applications based on a graphical user interface (GUI) using the Eclipse Standard Widget Toolkit (www.eclipse.org/swt/). The recorder tool monitors the main source of non-determinism in GUI based applications which are user inputs (keystrokes and mouse events). Additionally the recorder tool monitors two application triggers of error report: unhandled exceptions and explicit user requests for assistance.

Error reports are transferred to the FastFix server using the FastFix Communication System. Once at the server, error reports are stored and can be replayed at any time.

2.1 Code Metrics

Since the platform is based on the OSGi framework, the recorder/replayer tool is organized as a set of bundles hosted at https://repository.fastfixproject.eu/svn/fastfix (authenticated access). The bundles are:

- eu.fastfix.client.faultReplication.sensing: a context sensor which monitors the SWT application log for unhandled exceptions (4 classes, 13 methods, 157 lines of code [LOC]).
- eu.fastfix.client.faultReplication.userReport: Eclipse plugin allowing application users to request assistance (2 classes, 7 methods, 129 LOC).
- eu.fastfix.client.faultReplication: this bundle performs application execution information processing on the client side (7 classes, 33 methods, 321 LOC).
- eu.fastfix.client.faultReplication.guiRecorder: this bundle records relevant GUI events on a FastFix enabled application (4 classes, 19 methods,
264 LOC).

- `eu.fastfix.common.faultReplication.gui`: common utility classes for the GUI recorder and replayer (4 classes, 19 methods, 227 LOC).
- `eu.fastfix.server.faultReplication.guiReplayer`: GUI execution replayer (6 classes, 24 methods, 370 LOC).
3 Execution Recorder Tool

Any OSGi/SWT application that uses FastFix’s fault replication must use components (OSGi bundles in the present case) that perform two functionalities: recording the application’s execution and triggering an error report. The first aspect is assured by the eu.fastfix.client.faultReplication.guiRecorder bundle. From the beginning of the execution of any SWT/FastFix application all GUI events are recorded in a log file. The second part, error reporting can be automatic or explicit.

The automatic error reporting is performed by the eu.fastfix.client.faultReplication.sensing bundle which detects runtime errors in the form of unhandled exceptions. In SWT application unhandled exceptions are logged in an application log. Whenever this log monitoring sensor detects that an unhandled exception has been recorded it triggers an error report.

If error reporting is to be done explicitly by the user, the application must include the eu.fastfix.client.faultReplication.userReport bundle in its OSGi execution configuration. This will insert an option called “Request Assistance” in the “Help” menu. By selecting this option users will trigger an error report. The report will include a user written assistance request (as well as the execution log describing what the user did before requesting assistance).

The error report (unhandled exception or assistance request) is communicated within the platform by broadcasting an “Unhandled Exception” event on the FastFix Context Bus.

This context event is detected by the Client Fault Replication System, which then processes the information gathered by the application and its sensors and prepares the report to be sent to the FastFix server side. Future versions of the Client Fault Replication System will perform user data anonymization. For now, all input data and GUI events are sent unmodified to the FastFix server. The error report is sent by the Client Error Report Generation System described in D3.5.
4 Execution Replayer Tool

The FastFix execution replayer tool replays executions recorded at the FastFix server site. In practice it is an OSGi bundle (eu.fastfix.server.faultReplication.guiReplayer) that must added to the run configuration of any application for which errors are replayed.

Error reports are stored in the TRAC ticketing system. Additional files related to each of the TRAC tickets are stored in the FastFix server’s file system.

When the maintenance engineers using the FastFix server replay the error associated to a particular TRAC ticket, the corresponding application should be started explicitly and the replayer OSGi bundle (eu.fastfix.server.faultReplication.guiReplayer) is automatically started (see Figure 1 top right corner).

![Screenshot of the replay of a FastFix-enabled application.](image)

The engineers should insert the error report’s ticket id (obtained from the ticketing system’s interface) and the exact execution which triggered that ticket will be run. Future versions will allow users to browse the tickets in the ticketing system and select one for replay. The
execution replay can happen at the same speed as it originally happened on the client or it can be replayed with a user-defined time interval between replay steps. The replayer makes it possible to fastforward the replay to a particular point in time. Additionally, the replayer displays context events that happened at the client device during the execution of the replayed application such as mouse clicks and key presses (see for example Figure 1 bottom right corner). Future releases will also display additional context events recorded together with the error report’s execution trace.
5 Running FastFix Record-Replay

Currently the FastFix platform is run as a set of OSGi components. The best way to test the components described in this document is in the context of the FastFix development environment.

5.1 Running FastFix Record-Replay as an End-User

A. On the FastFix Server Machine:

Requirements:
- MySQL installed (e.g. MAMP or LAMP)
- Java 1.5 installed
- SSH server / daemon running.
- The SWT OSGi application you want to run within FastFix.

Installation:
1. Download FastFix Server from https://repository.fastfixproject.eu/svn/fastfix/Software/trunk/deployment/fastfix_server/fastfix-server-mac.ZIP (Choose your platform)
2. Unzip the downloaded file into a directory “fastfixserver”.
3. Start the FastFix Server by double-clicking on the JAR file in the “fastfixserver” directory.
4. Open the preferences menu (Menu “FastFix Server -> Preferences…”)
5. FastFix has to connect to the MySQL instance. Please make sure that the specified host, port, and credentials match your MySQL installation.
6. Download MOSKitt replayer bundles from https://repository.fastfixproject.eu/svn/fastfix/Software/trunk/deployment/error_replay/replay.zip. These bundles should be added to the runtime of the OSGi application that you want to run within FastFix.

B. On the “Client Machine”:

Requirements:
- Java 1.5 installed
- The SWT OSGi application you want to run within FastFix.

Installation:
1. **Download FastFix Client** from
   [https://repository.fastfixproject.eu/svn/fastfix/Software/trunk/deployment/fastfix_client/fastfix-client-mac.zip](https://repository.fastfixproject.eu/svn/fastfix/Software/trunk/deployment/fastfix_client/fastfix-client-mac.zip) (Choose your platform)
2. **Unzip** the downloaded file into a directory “fastfixclient”.
3. **Start** the FastFix Client by double-clicking on the JAR file in the “fastfixclient” directory.
4. **Enable** the FastFix Client by clicking the first icon, marked with a red circle in Figure 2.

   ![Figure 2: FastFix Client UI](image)

5. **Start your application** with the required FastFix bundles for replaying:
   - eu.fastfix.client.faultReplication.guiRecorder,
   - eu.fastfix.client.faultReplication.sensing,
   - eu.fastfix.client.faultReplication and
   - eu.fastfix.common.communication

**Usage:**
1. **Switch** to the FastFix Client.
2. Open the **preferences menu** (Menu “FastFix Client -> Preferences…”)
3. FastFix client has to send application execution traces to the “Server Machine” 
   via SCP. Please make sure that the specified **host, username, and password** 
   match the SSH / SCP parameters on the “Server Machine”.
4. If step 3 does not work (possible on some machines):
   a. Close the FastFix Client
   b. Change into its main directory ("fastfixclient"), and there change into the 
      plugins folder
   c. Change into the "eu.fastfix.common.communication_0.0.1.SNAPSHOT" 
      folder.
   d. Open the **CommunicationProperties** file, and adapt the FastFix server 
      IP address and port. Save and close the file.
   e. Go one folder up again.
f. Change into the “eu.fastfix.client.error.report.generation_0.0.1.SNAPSHOT” folder.
g. Open the “report.properties” file, and adapt URL, user, password, and folder of where the execution trace will go. Save and close the file.
h. Restart the FastFix client.

5. In the UI you should see two sensors as shown in Figure 3.
6. Click on the “start all sensors” symbol (marked with a red circle in Figure 3).

Figure 3: FastFix Client with MOSKitt Sensors

7. FastFix is now up and running.
8. To test, just work normally with your application.

9. If FastFix detects an unhandled exception, you can find the corresponding error report at your TRAC server.
10. To replay the application execution that lead to this exception, switch to the “Server Machine”.
11. Start your application on the “Server Machine”
12. You should see two new replayer windows, as shown in Figure 4
13. Do not use the application features now.
14. Enter the ticket number of the ticket to replay in the replayer menu.
15. Hit Replay to replay the error.
5.2 Running FastFix Record-Replay as a Developer

5.2.1 Infrastructure for FastFix Development

The following software is required to develop FastFix components:


- [http://m2eclipse.sonatype.org/sites/m2e/](http://m2eclipse.sonatype.org/sites/m2e/)

5.2.2 Build and Run FastFix

5.2.2.1 Build

For each platform, client and server, there is a dedicated Maven Project called "parent". This is needed in order to build all required bundles for the client or the server automatically with Maven. The server parent bundle is called eu.fastfix.server.parent and the client parent project eu.fastfix.client.parent. To build the client bundles

1. Right-click on the client parent project and select "Run As... -> Maven Install".

2. Wait for Maven to build the packages and generate the binaries and Manifest.MF files. As a result, you should get something similar to this:
3. Refresh your package explorer by pressing F5 or right-click on it and press "Refresh" in order to prevent "Out of sync" while running the bundles.

Perform analogous steps for the server bundles.

5.2.2.2 Run existing bundles

To run the current bundles

4. Select one of the projects by right-clicking on it
5. Choose "Run As -> Run Configurations...".
6. Double-Click on "OSGi Framework" item from the list placed left. This will create a new configuration file for running the bundles.
7. In the "Arguments tab", replace the proposed "Program arguments" with
   1. 

   

   

   

8. Press "Run"
9. In the "Console" view, type "ss fastfix" and press enter in order to display all FastFix bundles. If any of the FastFix bundles has "Resolved" state, start it by typing "start xx" where "xx" is the number of the plugin which is not activated.

One only has to create a new configuration and specify arguments once. Later just hit on "Run as OSGi Framework".