

Title	Audit and log
Product	BLT_TREJB2, BLT_TRCS2
Last update	08/2/2010

## 1 Purpose/Target

The main purpose of this document is to describe a new feature to be added to the transformation engines for the business logic tier, which will allow generating log files with audit information about the execution of services and queries. This document also outlines the behavior of the code that will support this new feature.

## 2 Description

### 2.1 Introduction

The current code obtained from a conceptual model using **OLIVANOVA** transformation engines does not offer the possibility of obtaining log files to store some useful information to perform audit tasks.

This new feature to be added to the generated code, will offer the possibility to generate these files, using specific libraries and classes that help to build the message and generate said files.

### 2.2 Proposal

In order to perform audit tasks it is often helpful to have feedback about the execution of services and queries. This new feature aims at providing the minimum code-level infrastructure to support the generation of log files with audit information about the execution of services and queries.

Such information may depend on the connected agent and the services/queries requested. As a default this feature will produce log information whenever any agent requests the execution of any service or whenever any agent requests the execution of a query.

Developers can customize the generated code to limit this feature to the agents, services and queries (in terms of queried attributes/relations) of their choice.

#### 2.2.1 Services

For each service in the conceptual model may be subject to generating audit info whenever an agent requests its execution. By default, the generated code will log audit info for any agent executing any service. Information about how to customize this default behavior will be provided in a different document.

When an agent requests the execution of a service the code corresponding to this new feature will check whether it has to log audit info (This is determined by the class the agent belongs to and the service said agent requests to execute). If so, it will add a new entry to the log file with the following information: identification of the agent requesting the execution of the service, name of the service and name of the class owning said service.

Should the execution of the service fail, the code corresponding to this new feature will add a new entry to the log file indicating said failure with the following information: identification of the agent requesting the execution of the service that failed, name of the

service that failed and name of the class owning said service.

This behavior applies irrespectively of whether the service is atomic (event) or molecular (local/global transaction/operation). It is worth noticing that log information will be produced when an agent requests the execution of a molecular service, but no log information will be produced for the execution of the services comprised in said molecular service.

Non-transactional molecular services (local/global operations) do not, per se, rollback any changes made to the system upon the failure of any of the transactional services comprised therein (it is the responsibility of said transactional services to rollback in case of failure). Therefore, failures in any of the transactional services comprised in a non-transactional molecular service are masked by the latter and thus, produce no log information. However, log information will be logged in case something related to the execution of the non-transactional molecular service fails (e.g. a precondition not fulfilled, a runtime error in evaluating a guard).

### **2.2.2 Queries**

Querying the population of any class in the conceptual model may be subject to generating audit info whenever an agent requests said query. By default, the generated code will log audit info for any agent querying the population of a class, regardless of the attributes requested in said query. Information about how to customize this default behavior will be provided in a different document.

When an agent queries the population of a class the code corresponding to this new feature will check whether it has to log audit info (This is determined by the class the agent belongs to, the class whose population is to be queried and the attributes said agent wants to get as a result of the query). If so, it will add a new entry to the log file with the following information: identification of the agent requesting the query, the name of the class whose population is to be queried, and the names of the attributes that said agent requested to query and for which audit info is to be logged.

Should the execution of the query fail, the code corresponding to this new feature will add a new entry to the log file indicating said failure with the following information: identification of the agent requesting the query, the name of the class whose population is to be queried, and the names of the attributes that said agent requested to query and for which audit info is to be logged.

Information about the attributes requested as part of the query is logged independently of the visibility that the requesting agent has. This means that if a given agent is querying the population of a given class and requests attribute "A", even if said agent has no visibility of said "A" attribute, there will be audit info indicating that "A" was requested as part of the query. This will allow to detect situations in which agents are querying information they are not supposed to have access to.

### 3 Remarks

This new feature will be offered as an option in **OLIVANOVA** STAR Client for the following profiles:

- BLT\_TRCS2: Business Logic Transformation Engine Transactional C# .Net 2.0.
- BLT\_TREJB2: Business Logic Transformation Engine Transactional EJB2 Architecture JAVA Platform.

The following will have to be taken into account when dealing with this new option:

- The default value for this new option selected in **OLIVANOVA** STAR Client will be false (by default the generated code will *\*not\** include support for logging audit info).
- If the value for this new option is set to true in **OLIVANOVA** STAR Client, the generated code will be configured to log audit info for the execution of any service by any agent and for the execution of any query on any attribute of any class of the model by any agent. It will be up to the developer to customize this default behavior. A "how-to" document will be provided in order to explain how to perform this customization.