

## **Oracle Agile Engineering Data Management**

Batch Client for Agile

Release e6.2.1.0

**E69110-01**

August 2017

Oracle Agile Engineering Data Management/Batch Client for Agile, Release e6.2.1.0

E69110-01

Copyright © 1995, 2017, Oracle and/or its affiliates. All rights reserved.

Primary Author:

Contributing Author:

Contributor:

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

---

---

# Contents

<b>Preface</b> .....	v
Audience .....	v
Documentation Accessibility .....	v
Related Documents .....	v
Conventions .....	v
 <b>1 Introduction</b>	
 <b>2 Installing and Configuring the Batch Client</b>	
Installing the Agile e6.2.1.0 Batch Client .....	2-1
Configuring the Agile e6.2.1.0 Batch Client .....	2-2
 <b>3 Batch Client Settings</b>	
Batch Client Solution .....	3-1
Java Wrapper Settings .....	3-1
Configurations .....	3-1
Java Application Settings .....	3-1
MS Windows Service Settings .....	3-1
Agile e6.2.1.0 Batch Client Properties .....	3-2
Secure User Credentials .....	3-3
Enable the Old Batch Solution .....	3-4
Configuring the Batch Client for Executing a Simple LogiView Procedure .....	3-4
 <b>4 Use Cases</b>	
Scenario Mode .....	4-1
Recommended Design of a Batch Control .....	4-1
Time-Out Value Information .....	4-3
Run-time Process (Batch Job Loop) .....	4-4
BatchExample/setRunning .....	4-5
BatchExample/shouldShutdown .....	4-5
BatchExample/executeOperation .....	4-5
BatchExample/setStopped .....	4-5
ECI Server Mode .....	4-6
Wrapper Configuration Settings .....	4-6

ECI Server Settings.....	4-7
Office Suite PDF Service.....	4-7
AutoVue Offline Metafile Cache Service.....	4-7

---

---

# Preface

Agile PLM is a comprehensive enterprise PLM solution for managing your product value chain.

## Audience

This document is intended for administrators and users of the Agile PLM products.

## Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at  
<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

### Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit  
<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit  
<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

## Related Documents

Oracle's Agile PLM documentation set includes Adobe® Acrobat PDF files. The Oracle Technology Network (OTN) website  
<http://www.oracle.com/technetwork/documentation/agile-085940.html> contains the latest versions of the Agile PLM PDF files. You can view or download these manuals from the Web site, or you can ask your Agile administrator if there is an Agile PLM Documentation folder available on your network from which you can access the Agile PLM documentation (PDF) files.

## Conventions

The following text conventions are used in this document:

Convention	Meaning
<b>boldface</b>	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.

Convention	Meaning
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

---

# Introduction

---

**Note:** The "old" EDB\_BATCH mechanism of the EDM server is not recommended anymore, and the Batch Client replaces the "old" batch solution of the EDM server.

---

The Batch Client runs a batch scenario which is implemented in Groovy to control the batch use case. The most batch use cases are designed as a loop, which uses a job table to determine if a job is present or if the process should wait for a new job.

The Agile e6.2.1.0 Batch Client supports two different modes:

1. Scenario Mode

The Batch Client starts a defined scenario which interacts with the EDM Server to execute the batch use case.

2. ECI Server Mode

The Batch Client provides an ECI tunnel to the EDM server and standard features like file access to the File Server.

For more details, refer to the [Chapter 4, "Use Cases."](#)

---

**Note:** For more information regarding the Batch Client architecture, refer to the section, Application Server Components of the *Architecture Guide for Agile e6.2.1.0*.

---

---

**Note:** The Agile e6.2.1.0 Batch Client is available for all server platforms of the Agile e6.2.1.0 release. For more details, refer to the *Oracle Agile Engineering Data Management - Platform Support for Agile e6.2.1.0*

---





---

## Installing and Configuring the Batch Client

---

---

**Note:** The Agile e6.2.1.0 Batch Client has no Installer software.

---

### Installing the Agile e6.2.1.0 Batch Client

The Agile e6.2.1.0 Batch Client is delivered as a ZIP file.

The ZIP file is contained in the batchclient.zip installation package located in the *packages* folder of the installation medium.

1. Copy the Batch Client installation package (batchclient.zip) from the *packages* folder of the installation medium into a temporary directory

Example:

```
copy y:\packages\batchclient.zip c:\temp
```

2. Extract the installation package.

This unpacks the Agile e6.2.1.0 Batch Client ZIP archive into the *axalant/bin/java* sub directory of your temporary directory.

Example:

```
unzip c:\temp\batchclient.zip -d c:\temp
```

3. Extract the Agile e6.2.1.0 Batch Client bundle into the installation directory.

Example: C:\Program Files.

The Agile e6.2.1.0 Batch Client files are unpacked into the *batchclient* directory which is automatically created.

Example:

```
unzip c:\temp\axalant\bin\java\batchclient.zip -d "c:\Program Files\Agile_e6"
```

The Agile e6.2.1.0 Batch Client bundle contains the following directories:

---

#### Directory paths

---

Batch Client  
Root

(C:\Program Files\Agile\_  
e6\Batchclient)

axalant

bin

Directory paths		
	x86_64-linux-ol7	Oracle Linux FMS-Client binary
	ia64w-hp-hpux11.31	HPUX FMS-Client binary
	x64-ms-nt6.1	Windows 64 bit FMS-Client binaries
	java	Java archives
	ppc64-ibm-aix7.1	IBM AIX FMS-Client binary
	sparc64-sun-solaris11	Oracle Solaris FMS-Client binary
	cmd	Windows scripts directory
	dmp	Example loader files
	ini	Configuration files
	scripts	UNIX scripts directory
examples		Example batch files
ext		
	bin	
	x86_64-linux-ol7	external Oracle Linux binaries
	ia64w-hp-hpux11.31	external HP-UX binaries
	x64-ms-nt6.1	external Windows 64 bit binaries
	java	external Java archives
	ppc64-ibm-aix7.1	external IBM AIX binaries
	sparc64-sun-solaris11	external Oracle Solaris binaries
	tmp	Log files

## Configuring the Agile e6.2.1.0 Batch Client

The Java environment has to be set to Java 8 64-bit JRE version.

- On Windows, adapt the *BatchClient.cmd* shell script:

Example:

```
set JAVA_HOME="C:\Program Files\Java\jre1.8.0_<update_number>"
```

- On UNIX, adapt the Batch Client shell script:

Example:

```
JAVA_HOME=/usr/local/java/jre1.8.0_<update_number>
```

## Batch Client Settings

This chapter describes the several settings related to the Java Wrapper framework of the Agile e6.2.1.0 Batch Client in the Scenario mode and ECI mode.

### Batch Client Solution

The Agile e6.2.1.0 Batch Client uses the Java Wrapper as its run-time frame work.

### Java Wrapper Settings

The Java Wrapper needs a configuration file:

```
<installdir>\axalant\ini\batch_simple_wrapper.conf
```

The Java Wrapper supports the following shutdown mode:

- Simple type:

The Java application, controlled by the Java Wrapper, will be configured and simply killed at shutdown. This mode is used for the Java Daemon

### Configurations

Here is a short description of the settings which may be adapted to implement a custom batch solution.

### Java Application Settings

This section in the configuration defines the controlled Java application and its command line parameters.

```
# Application parameters. Add parameters as needed starting from 1.
wrapper.app.parameter.1=com.agile.testclient.BatchClientApp
wrapper.app.parameter.2=-p%EP_ROOT%/examples/LgvLoop.properties
wrapper.app.parameter.3=-i%EP_ROOT%/examples/LgvLoop.groovy
wrapper.app.parameter.4=-T%EP_ROOT%/axalant/ini/testclient.properties
```

### MS Windows Service Settings

This section within the configuration file defines the MS Windows Service settings. Here, the name of the server can be defined and some extended settings for services.

```
*****
# Wrapper NT Service Properties
*****
# WARNING - Do not modify any of these properties when an application using this
```

```

configuration file has been installed as a service.
# Please uninstall the service before modifying this section. The service can then
be reinstalled.
# Name of the service
wrapper.ntservice.name=AgilePLM_Batch_Service
# Display name of the service
wrapper.ntservice.displayname=AgilePLM_Batch_Service
# Description of the service
wrapper.ntservice.description=Batch Service for Agile PLM
# Service dependencies. Add dependencies as needed starting from 1
wrapper.ntservice.dependency.1=
# Mode in which the service is installed. AUTO_START or DEMAND_START
wrapper.ntservice.starttype=AUTO_START
# Allow the service to interact with the desktop.
wrapper.ntservice.interactive=false

```

### Agile e6.2.1.0 Batch Client Properties

The used Agile e6.2.1.0 Batch Client scenario and the corresponding property file are defined as command line parameters in the Java Wrapper configuration file as you can see above.

This configuration file has to be adapted to your environment.

---

**Note:** The Batch Client bundle provides an example (located in directory C:\Program Files\Agile\_e6\batchclient\examples) which has to be adapted to your environment.

---

host=<PLM host>	Hostname of the Agile e6 server
port=<PLM port>	Port number of the Java Daemon
env=<PLM env>	Application name to connect to
url=<PLM-API URL>	Defines the URL for the HTTP Proxy of the Agile e6 server. If you want to use the URL, you need to uncomment the line in the configuration file.  Example: url=http://plm.example.com:7101/plm-api-axis/services
encoding=UTF-8	Default encoding for the ECI connection
scope=BATCH	Scope for the Java Daemon. This should always be BATCH for batch applications
varenv.EP_DDM_SITE	The DDM Site
url	The PLM-API URL. If connection to e6 should be made via PLM-API the HTTPS URL of the PLM-API must be used. The Weblogic server must also be configured to use HTTPS. For more information, see the <i>Architecture Guide</i> , <i>Administration Guide</i> and the <i>Online Help</i> .
varenv.ep_root=<install dir>/Agile_e6/BatchClient	Directories
varenv.axalant_root=<install dir>/Agile_e6/BatchClient/axalant	

---

varenv.EP_MACH=<client ep_mach>	Platform
Security.KeyStoreFile=file://<install dir>/Agile_e6/BatchClient/init/wallet/private/batch/cwallet.sso	Security settings
Security.KeyAlias=orakey	
client1=<PLM user>,<PLM password>,com.agile.EciServer	PLM Client
varenv.EciServer.StartupImmediately=true	
varenv.EciServer.ConnectionDelay=100	Delay between two incoming ECI connection requests in milliseconds
varenv.EP_DDM_SITE=<DDM site>	Sets the DFM site for the Batch Client.
varenv.HTTP_FMS_MODE=false	Defines if the HTTP FMS Client or the native FMS Client must be used. Possible values: true / false Default: false true = The HTTP FMS Client will be used. false = The native FMS Client will be used.

---

The security setting needs to be adapted in the Batch Client properties. The SecurityKeyStoreFile has to point to the Oracle wallet of the Batch Client installation.

In the next section you can find instructions on how to create an Oracle wallet for your batch client installation.

#### Example Windows:

```
Security.KeyStoreFile=file://C:/Program Files/Agile_e6/BatchClient/init/wallet/private/batch/cwallet.sso
```

#### Example UNIX:

```
Security.KeyStoreFile=file:///app/plm/BatchClient/init/wallet/private/batch/cwallet.sso
```

---

---

**Note:** The Batch mode can be set in the property files LGVCall.properties, LgvLoop.properties, OfsPdf.properties, and vuelink.properties.

The eciserver mode should always be BATCH.

When no mode is set in the property files, the default value is BATCH.

---

---

## Secure User Credentials

The User credentials are stored in the Agile e6.2.1.0 Batch Client properties file.

The Batch Client installation needs to create a new installation specific Oracle wallet which needs to be protected against unauthorized file access. For more information refer to the Agile EDM Security Guide Chapter Wallets > Batch Client.

## Enable the Old Batch Solution

---

**Note:** The "old" batch solution of the EDM server is still available, but due to security risks this solution is not recommended.

---

To enable the EDB\_BATCH mechanism, the security level has to be decreased to *unrestricted*.

The security level can be configured in the *<environment>.xml* file located in the *<ep\_root>/init* directory.

Example:

```
<IPC
AbsEciUrl="eci://localhost:19997" SecurityLevel="unrestricted"
TicketTimeout="600">
</IPC>
```

---

**Note:** The security level setting also controls the security settings of the ECI connection. A client needs no authentication to connect to an *unrestricted* ECI server. Please ensure that this setting is not effective for EDM servers, which allow to be contacted via ECI.

---

## Configuring the Batch Client for Executing a Simple LogiView Procedure

The standard Agile e6.2.1.0 Batch Client bundle is pre-configured for executing a LogiView Start/Stop scenario and cannot be used out of the box for a starting a simple LogiView scenario. Especially the wrapper configuration file needs a different wrapper Java MainClass and different wrapper application parameters.

The Agile e6.2.1.0 Batch Client requires the following configuration files / start scripts:

---

**Note:** the following files / start scripts can all be renamed!

---

```
<ep_root>\axalant\ini\batch_wrapper.conf
<ep_root>\axalant\ini\testclient.properties
<ep_root>\examples\LGVCall.properties
<ep_root>\examples\LGVCall.groovy
<ep_root>\axalant\cmd\batchclient.cmd (Windows)
<ep_root>/axalant/scripts/batchclient (Unix)
```

To enable the Batch Client for executing a simple LogiView procedure, the Agile e6.2.1.0 Batch Client bundle is delivered with pre-configured examples which call a simple LogiView (LGV) procedure. The example files are located in the installation folder as follows:

Batch command script	axalant/cmd/Simple.cmd
Wrapper configuration	axalant/ini/batch_simple_wrapper.conf
Batch scenario	axalant/examples/LGVCall.groovy
Batch configuration	axalant/examples/LGVCall.properties

The Agile e6.2.1.0 Batch Client supports two modes:

1. Scenario Mode:

To implement the Batch Use Cases.

2. ECI Server Mode:

To provide an ECI tunnel and client access like file server access.

## Scenario Mode

The scenario mode runs a Groovy script which implements the batch process.

Information about how to design and write a script to implement a Batch use case can be found in this chapter.

## Recommended Design of a Batch Control

---

---

**Note:** With the Agile e6.2.1.0 Batch Client bundle, a set of batch control scripts are available in the examples folder. The LogiView procedures of the examples are available as loader files. In this example the synchronization of the run-time process and the shutdown process is carried out by using an EDM configuration parameter within the database. But you can use any information source which is accessible by both processes, for example, a shutdown job in the job table, a local file, a record in the database, etc.

---

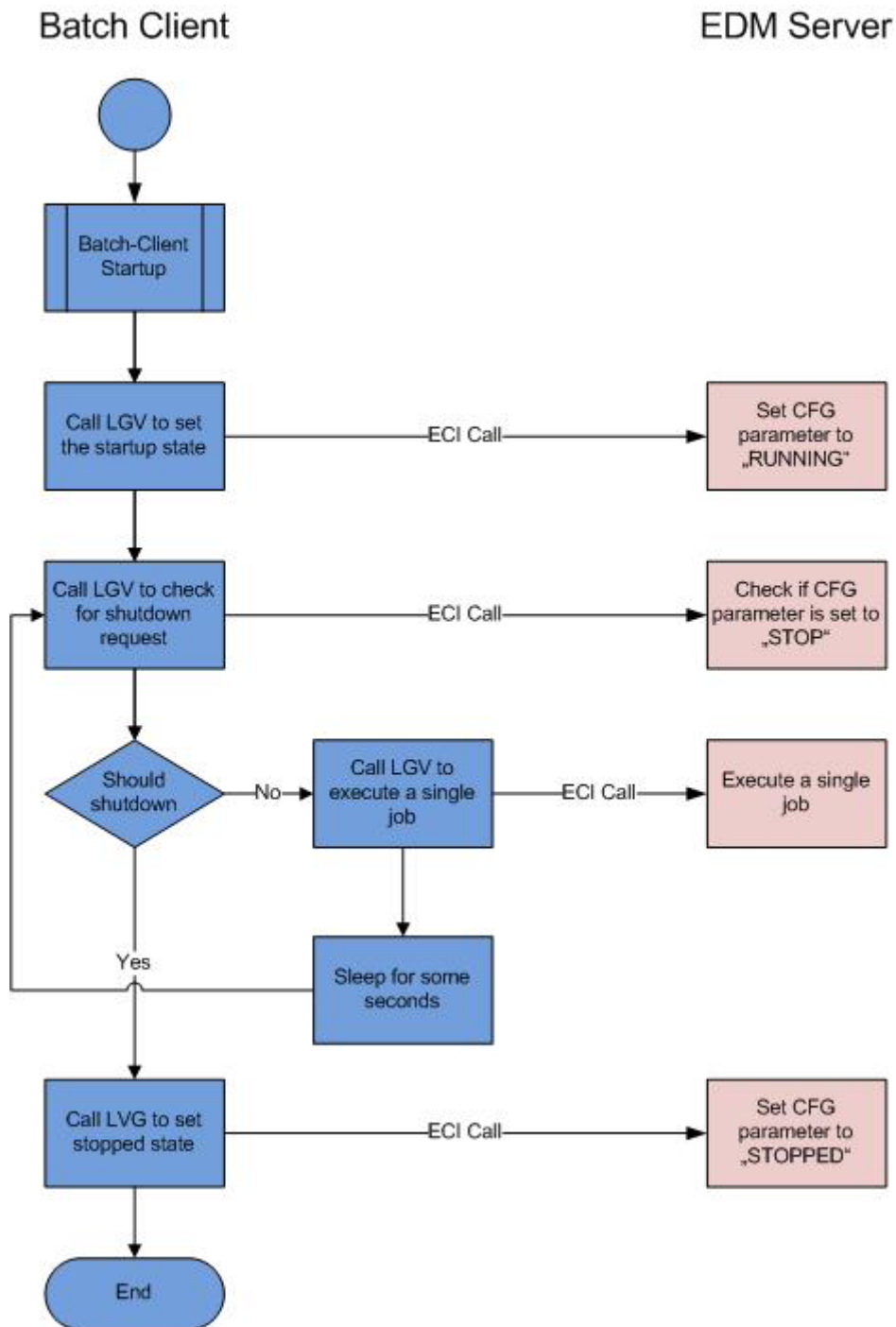
---

The Agile e6.2.1.0 Batch Client bundle comes with a set of batch control scripts. These are in the examples folder.

The LgvLoop.groovy script can be used as a template for each job table based batch use case. The script uses a set of LogiView procedures to check, control, and execute the batch functionality. The Agile e6.2.1.0 Batch Client provides a run-time process and a shutdown process. In the provided template, the Agile e6.2.1.0 Batch Client LogiView backbone uses a configuration parameter to report the current state of the batch process and to control the batch process.

The following flowchart shows a command flow of the run-time process.

## Batch Job Loop



The Java Wrapper executes the run-time process during start-up. The run-time process sets the current state of the batch job by using the corresponding LogiView procedures, and executes the batch job with the central LogiView procedure. To detect a shutdown request, the run-time progress checks if a shutdown request is available by calling a special LogiView procedure.

In this example, there are several LogiView procedures to report the current state of the run-time process:



- `setRunning`  
Called in the start-up phase to report that the batch job is running.
  - `shouldShutdown`  
Detects a shutdown request.
  - `executeOperation`  
Executes the batch job.
1. The run-time process calls `setRunning` during the start-up.
  2. The run-time process checks if there is a shutdown request available by calling `shouldShutdown`.
  3. If not, the `executeOperation` LogiView procedure is called to execute a batch job.  
If a shutdown request is available, the run-time process sets the "stopped" state by calling `setStopped` and ends the run-time scenario.

The shutdown process uses the following LogiView procedures:

- `shutdown`  
Creates a shutdown request.
- `isStopped`  
Detects that the run-time process has been stopped.

---

**Note:** The shutdown process waits until the run-time process has been stopped before the shutdown scenario ends.

---



---

**Note:** The Java Wrapper waits until the shutdown process ends or until the shutdown timeout has been reached. The shutdown timeout can be configured in the Java Wrapper configuration file.

---

## Time-Out Value Information

`wrapper.jvm_exit.timeout`

Defines the time in seconds that is allowed between the time JVM reports that the JVM is stopped and the time that the JVM process actually terminates.

- Default = 15 seconds
- No time out = 0 seconds

In a normal operation, the Java side of the Wrapper executes the file `System.exit` when it has completed its JVM shutdown cycle and is ready to exit. When this timeout is triggered, a message like the following is logged.

```
wrapper | Shutdown failed: Timed out waiting for the JVM to terminate.
wrapper | Java Virtual Machine did not exit on request, terminated
```

If the application has registered its own shutdown hook, which takes some time to complete, you could experience timeouts waiting for the JVM process to terminate. To avoid this problem, it may be necessary to extend the timeout to give the application's shutdown hook time to execute to completion. Be aware that, as a rule, shutdown hooks should always complete almost instantly.

Example: `wrapper.jvm_exit.timeout=5`

---

**Note:** For further information about the wrapper.jvm please refer to:  
<http://wrapper.tanukisoftware.org/doc/english/prop-jvm-exit-timeout.html>

---

### Run-time Process (Batch Job Loop)

The following is the Groovy script of the Batch Job Loop.

```
/**
 * Batchclient example script to demonstrate the basic implementation of a batch
 * service.
 * $Id: LgvLoop.groovy,v 34.2 2009/06/21 15:51:21 schwera1 Exp $
 */
package com.agile.LgvLoop;
import com.agile.testclient.*;
/**
 * Each scenario needs to extend the Scenario class.
 * The Scenario class provides the execution environment.
 */
public class LgvLoop extends Scenario {
/**
 * The run() method is called by the framework to start the scenario.
 */
public void run() {
System.out.println("Starting BatchExample/LgvProd");
    getTestClient().callNativeUsx("lgv_nosel_run", "BatchExample/setRunning");
    while(!shouldShutdown()) {
getTestClient().callNativeUsx("lgv_nosel_run", "BatchExample/executeOperation");
        sleepSeconds(10);
    }
getTestClient().callNativeUsx("lgv_nosel_run", "BatchExample/setStopped");
    System.out.println("Done BatchExample/LgvProd");
}
/**
 * Checks if the batch client should shutdown.
 * The method checks the DTV default "EDB-BATCH-CONTROL" to receive a shutdown
 * request.
 * @return true if the batch client should shutdown else false
 */
private boolean shouldShutdown() {
    int result = getTestClient().callNativeUsx("lgv_nosel_
run", "BatchExample/shouldShutdown");
    return result == 1;
}
}
```

The script uses the following methods to implement the run-time process:

- `callNativeUsx`  
Calls LogiView procedures.
- `sleepSeconds`  
Sleeps for some seconds to wait.
- `println`  
Print information into the log file.

The script calls 4 LogiView procedures to set the current process state, to execute the batch job, and to check for shutdown requests. Here are the LogiView procedures used by the batch use case:

### BatchExample/setRunning

This LogiView procedure sets the batch process control variable (configuration parameter: EDB-BATCH-EXAMPLE-CONTROL) to the value "RUNNING".

```
EDB_BATCH_VALUE = "RUNNING"
update(EDB_BATCH_VALUE)
where("T_CFG_DAT.EDB_ID" = "EDB-BATCH-EXAMPLE-CONTROL")
exec_update()
```

- The configuration parameter EDB-BATCH-EXAMPLE-CONTROL  
Can be used to check if the Agile e6.2.1.0 Batch Client is online or not. If your batch process does not allow running more than one Agile e6.2.1.0 Batch Client, you can add another LogiView call to check the current state of the batch process.
- The LogiView procedure BatchExample/isRunning  
Checks if there is another batch process already active. Thus, starting a new Agile e6.2.1.0 Batch Client can be prevented by checking the current state and implementing the batch script to exit in that case.

### BatchExample/shouldShutdown

LogiView procedure to check the configuration parameter EDB-BATCH-EXAMPLE-CONTROL for a shutdown request. You can also use different variables to reflect the current state and to control the batch process. LogiView reports a shutdown request if the configuration parameter EDB-BATCH-EXAMPLE-CONTROL is set to "STOP" or if the configuration parameter is not present in the system.

```
select(EDB_BATCH_VALUE)
where("T_CFG_DAT.EDB_ID" = "EDB-BATCH-EXAMPLE-CONTROL")
RES = exec_select(1)
if (RES == 1
    if (EDB_BATCH_VALUE == "STOP")
        exit()
    endif
else
    exit()
endif
```

### BatchExample/executeOperation

The procedure checks if there is a batch job available and executes that job, or it returns the message that the Agile e6.2.1.0 Batch Client can sleep for some seconds and try again later. You can also use these messages to trace the actions of the batch job in the client trace.

### BatchExample/setStopped

The batch process checks in every loop if a shutdown request is pending. Before the Agile e6.2.1.0 Batch Client shuts down, the LogiView procedure BatchExample/setStopped is called to set the current state of the batch process.

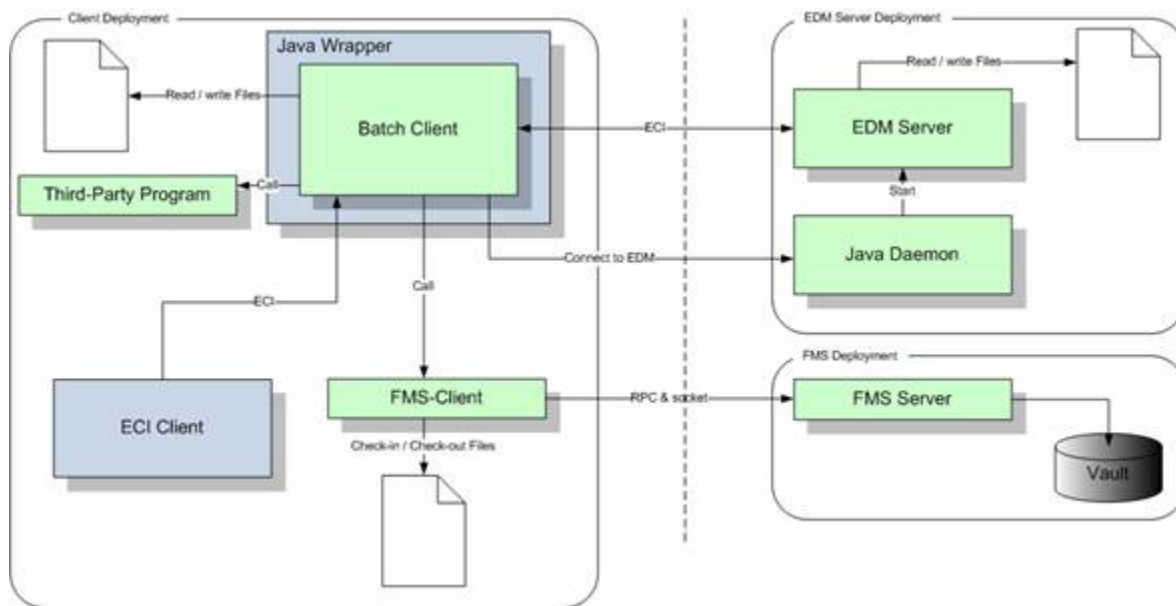
All LogiView sample procedures are available as loader files within the Agile e6.2.1.0 Batch Client bundle.

## ECI Server Mode

The Agile e6.2.1.0 Batch Client can be used as an ECI Server which tunnels the ECI calls of an ECI client to the EDM Server. Additionally to that tunnel functionality, the Agile e6.2.1.0 Batch Client provides some callables to start external programs or to interact with the FMS Client for file operations.

So a "simple" ECI client can be enabled to check in or check out files from the file vault without an interactive Java Client.

The following picture shows an overview of an Agile e6.2.1.0 Batch Client in ECI server mode.



The Agile e6.2.1.0 Batch Client in ECI Server mode uses the Java Wrapper to enable the service features, as it does in the Scenario mode.

The external ECI client can be, for example, a CAD integration which normally uses a Java Client to gain access to the file within a vault.

The Agile e6.2.1.0 Batch Client allows multi connections from several ECI clients.

To shutdown the Agile e6.2.1.0 Batch Client, the ECI client can call the `eci_stp_edb` ECI function.

### Wrapper Configuration Settings

To start the Agile e6.2.1.0 Batch Client in ECI Server mode, some wrapper configuration settings have to be changed as follows:

Java Environment:

The Java environment has to be set to a Java 8 or higher version.

- In Windows

Adapt the `eci_server.cmd` shell script for the Agile e6.2.1.0 Batch Client in ECI server mode.

Example: set JAVA\_HOME="C:\Program Files\Java\jre1.8.0\_<update\_number>"

- In UNIX

Adapt the ECI Server shell script.

Example: JAVA\_HOME=/usr/local/java/jjre1.8.0\_<update\_number>

## ECI Server Settings

The configuration of the Agile e6.2.1.0 Batch Client in ECI server mode is similar to the Agile e6.2.1.0 Batch Client in scenario mode with some small differences which are described here.

The ECI server mode does not need a scenario script, but the ECI server settings have to be configured.

Here are the additional settings within the Agile e6.2.1.0 Batch Client properties file:

varenv.EciServer.Port=4444	The port to use, a number > 1024, or 0 for any free port
varenv.EciServer.Encoding=UTF-8	The encoding to use for client connections. A client can switch the encoding by calling "eci_set_enc" after the connection has been established, or by specifying the desired encoding in the EciConParam object (in case of the Java-ECI).
varenv.EciServer.Secure=true	Check for authentication? Setting this to "false" will disable the password protection of the client. If set to "false", any program can execute PLM functions on behalf of the user's account as long as the client is connected to the PLM system. This should only be set to "false" if an e6 integration explicitly requires it.
varenv.EciServer.StartupImmediately=true	Startup mode. Controls the startup behavior of the ECI server. If set to false, incoming client requests are not answered until the connection to the Agile e6 server has been made. If set to true, the client may execute local ECI calls, but calls to the server are rejected.
varenv.EciServer.ConnectionDelay=100	Delay between two incoming ECI connection requests in milliseconds
varenv.EciServer.CallablePackage.1=<custom_callable_package>	Once the necessary wrapper configuration settings have been modified, you can use EciServer.cmd - c to start the Agile e6.2.1.0 Batch Client in console mode and check with the command eci_test if the Agile e6.2.1.0 Batch Client works properly.

## Office Suite PDF Service

For detailed information on the Office Suite PDF Service, refer to the *Administration Guide for Agile e6 > Office Suite - PDF Generator Installation*.

## AutoVue Offline Metafile Cache Service

For detailed information on the AutoVue Offline Metafile Cache Service, refer to the *AutoVue Integration Installation and Administration Guide for Agile e6*.

