

# How-To Guide - Enactor Logs

Document Type: External

Document Version: 1.1

Date: 01/09/2024

Author: John Wood

# Table of Contents

Table of Contents .....	2
Introduction.....	3
Overview.....	3
Enactor Logging .....	4
POS Logging .....	4
pos.log .....	4
common.log.....	4
PDP Server Logging.....	5
Listener.log .....	5
Common.log .....	5
Store Server Logging.....	6
Enactor Logs .....	6
Apache Tomcat Logs.....	7
Estate Manager Logging .....	8
Activity Log .....	8
Application Log .....	9
Application Process Log .....	12
Changing Logging Configuration.....	15
Class Logging.....	15
Process Logging .....	16
About This Document.....	17
Current Document Version information.....	17
Document Context .....	17
Document Inquiries .....	17
Document History .....	18

# Introduction

This guide will describe the numerous logs that are created by the Enactor suite of applications and will provide guidance for configuring logging for specific cases.

## Overview

This guide will cover the following:

- Description of the different types of logging in Enactor
- Locations of logs across different devices
- Configuration of log property files to include increasing and decreasing the level of detail included in each log and switching on logging for specific scenarios.

## Prior Training / Experience

You should be familiar with the following:

- Editing configuration files

# Enactor Logging

This section will be divided by device type and will cover the POS, the PDP Server, the Store Server, and the Estate Manager.

## POS Logging

POS logs are generated in the <Enactor home>/logs directory.

The two most important logs are pos.log and common.log

### pos.log

Pos.log is generated by the POS application and can be used for troubleshooting any issues seen on the POS UI (e.g., if a system error message is displayed on the POS, a corresponding log entry will be written to pos.log)

Configuration settings for pos.log can be found in <Enactor home>/pos/config/enactor-pos.log.properties

### common.log

Common.log is generated by the Enactor POS Services, which run as a Windows service. This log can be used for diagnosing issues with any background processing (e.g., receiving broadcasts or document processing)

Configuration settings for common.log can be found in <Enactor home>/pos/config/enactor.log.properties

## PDP Server Logging

PDP logs are generated in the <Enactor home>/logs directory.

The two most important logs are listener.log and common.log

### Listener.log

Listener.log is generated by the Enactor PDP Listener Windows service and can be used for troubleshooting any issues seen on the POS UI (e.g., if a system error message is displayed on a mobile client device, a corresponding log entry will be written to listener.log)

Configuration settings for listener.log can be found in <Enactor home>/pdpServer/config/enactor-pdp-server.log.properties

### Common.log

Common.log is generated by the Enactor PDP Services, which run as a Windows service. This log can be used for diagnosing issues with any background processing (e.g., receiving broadcasts or document processing)

Configuration settings for common.log can be found in <Enactor home>/pdpServer/config/enactor.log.properties

# Store Server Logging

## Enactor Logs

Numerous Enactor logs are generated on the Store Server.

Before looking at the logs, it's useful to have a basic understanding of how the Store Server application is structured.

When a Store Server is built, an instance of Apache Tomcat is installed and numerous web applications are deployed in four separate WAR files, as described below:

Web Core – Contains the services that should be deployed on every tier of the Enactor Estate

Web Reports – Responsible for generating reports on the Store Server

Web Store Management – Provides the Store Server UI

Web Store Processing – Provides processing services for the Store Server

Each of these web applications generates its own log – all logs are written to <Enactor home>\storeServer\logs.

### *Web-core.log*

Web-core.log is generated by the Enactor Web Core application and can be used for troubleshooting any issues with the Enactor core services.

Configuration settings for web-core.log can be found in <Enactor home>/storeServer/config/enactor-web-core.log.properties.

### *Web-reports.log*

Web-reports.log is generated by the Enactor Web Reports application and can be used for troubleshooting report issues.

Configuration settings for web-reports.log can be found in <Enactor home>/storeServer/config/enactor-web-reports.log.properties.

### *Web-store-management.log*

Web-store-management.log is generated by the Enactor Web Store Management application. This log can be used for diagnosing issues with the numerous applications contained within the Store Server web UI.

Configuration settings for web-store-management.log can be found in <Enactor home>/storeServer/config/enactor-web-store-management.log.properties.

### *Web-store-processing.log*

Web-store-processing.log is generated by the Enactor Web Store Processing application. This log can be used for diagnosing issues with background processing (e.g. receiving broadcasts or document processing).

Configuration settings for processing.log can be found in <Enactor home>/storeServer/config/enactor-web-store-processing.log.properties.

### Apache Tomcat Logs

In addition to the logs written by the Enactor application, numerous logs are written by the Apache Tomcat web server. Enactor configures Apache Tomcat so these logs are also written to <Enactor home>\storeServer\logs.

Further information about Tomcat logging can be found at the following link <https://tomcat.apache.org/tomcat-9.0-doc/logging.html>

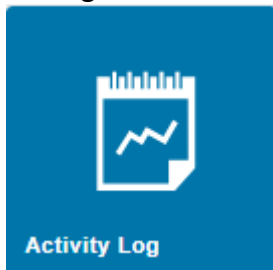
## Estate Manager Logging

The Enactor Estate Manager provides several logging applications within the UI. These can be found at the following path – Administration > Logs.



### Activity Log

The Activity Log provides a full audit trail of any changes made in the Enactor Estate Manager.



Entry Date	Entry Type	Entity Name	Location ID	User ID
18/11/2022, 11:16	Update	product	estatemanager	JOHN
18/11/2022, 11:16	Delete	productPrice	estatemanager	JOHN
18/11/2022, 11:15	Update	product	estatemanager	JOHN
18/11/2022, 11:15	Add	productPrice	estatemanager	JOHN
18/11/2022, 11:05	Add	product	estatemanager	JOHN
18/11/2022, 11:05	Add	productPrice	estatemanager	JOHN

To filter the entries listed enter values for the following fields and select **Apply Filters**:

Property	Description
Start Date	Beginning of a date range selection.
End Date	End of a date range selection.
Entry Type	Dropdown selection from a fixed list i.e. Data operations <b>Add</b> , <b>Update</b> , or <b>Delete</b> or logged Activities of <b>Other</b> types.
Entity Name	Selects for Activity pertaining to a specific Entity.
User ID	Selects for Activity conducted by a specific User.
Change Set	Dropdown list selection of defined Change Sets. Selects for Activity conducted in the specific Change Set.



### Activity Log Entry - General Tab View

Using the icons in the left column of the Select page, which are available as appropriate for each entry, the user can examine the individual Entries. For all entries the View option (🔍) provides access to an Activity Log entry viewing page.

For Entries that record Data Operations the View page provides access to before and after images of the entity as described following. The Activity Log entry also provides access to the Web Maintenance page for the relevant Entity to view either the pre-update or post-update state of the instance using the respective (←) or (→) Icons.

For **Other** operations the **Details Tab** for the entry provides details of the operation performed. In these cases, the View option (🔍) also provides a Raw XML Function viewing Tab as described following.

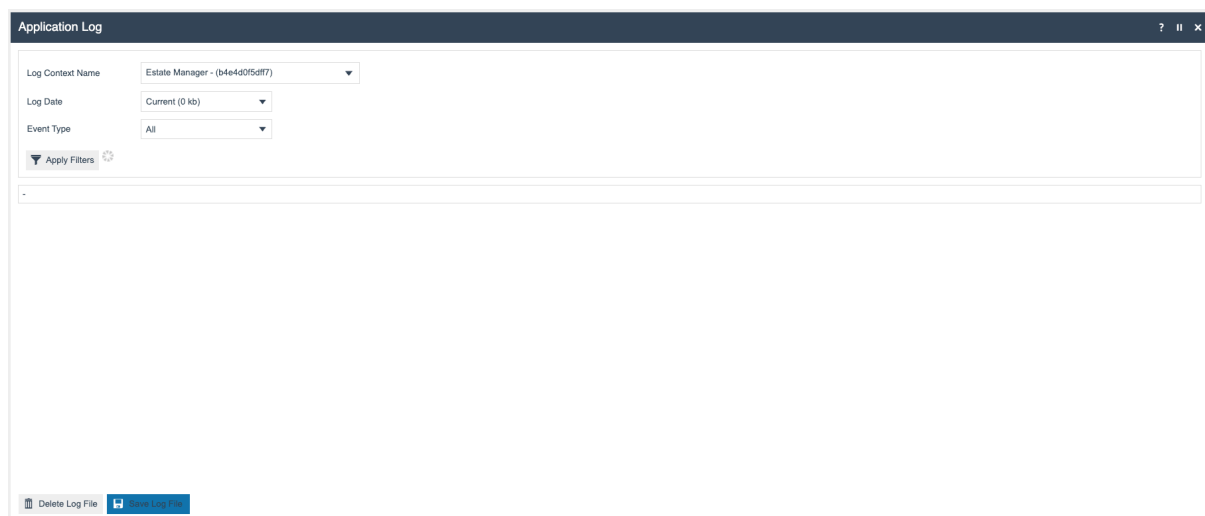
### Activity Log Entry - XML Before and After Action Tab View

For entries that record Data Operations (**Add**, **Update**, or **Delete**) the View page contains XML format before and after versions of the Entity instance in separate Tabs of the page. These XML representations of the instance may be extracted (by copy/paste) into text files for precise comparison or the pre-update image of an Update or deletion may be applied in the **File Import** facility of Web Maintenance to restore the instance (available at *Administration>Data Management*).

For **Other** operations the Function XML Tab of the Activity Log View page provides access to an XML representation of the Log Entry created by the Activity function, which may be a transaction, Message Queue entry or XML formatted logging details created by the activity function.

## Application Log

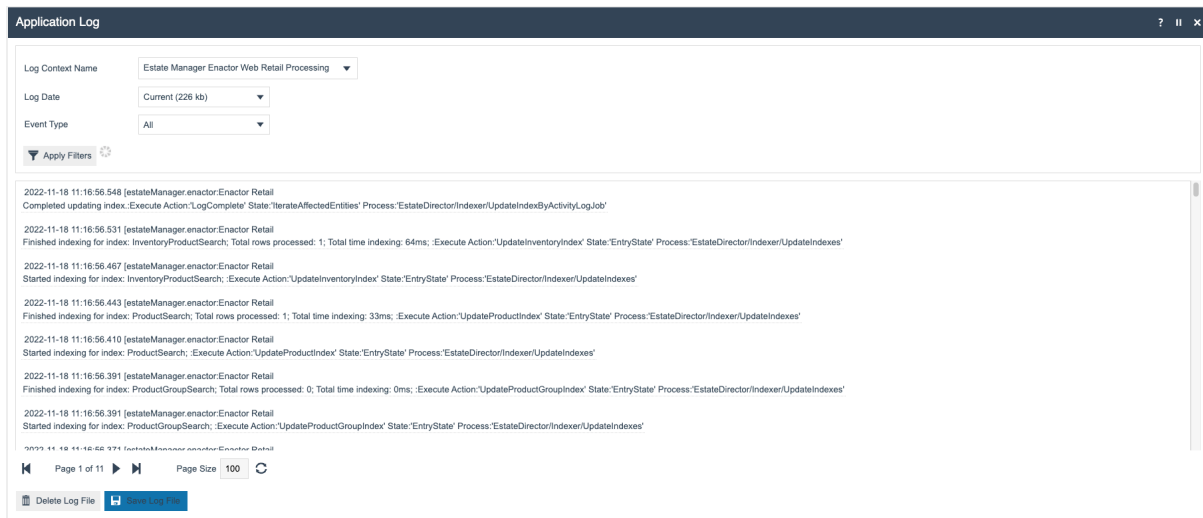
The Application Log provides a UI for all the logs generated by the various web applications that make up the Estate Manager.



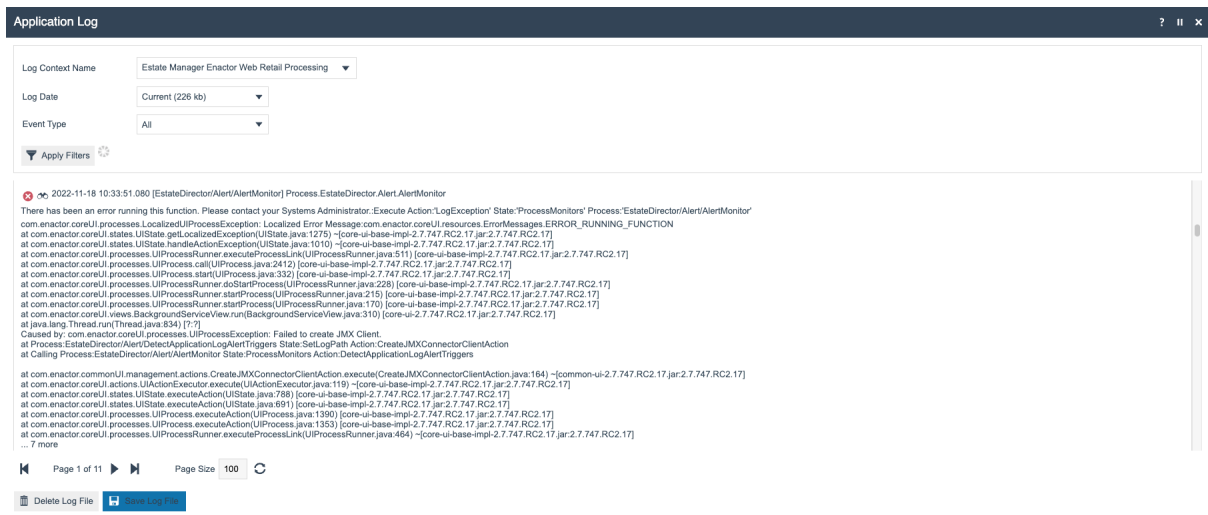
Selecting the Log Context Name drop-down gives a list of the available logs

- Card Authorisation Server - (d355da9a2c31)
- Card Authorisation Server Enactor Web Card Authorisation
- Customer Manager - (98599c32b6b6)
- Customer Manager Enactor Web Loyalty Services
- Customer Manager Enactor Web Message Service
- Customer Manager Rest API - Web Rest Api
- Estate Manager - (1cafdea7a65b)
- Estate Manager - (88de5ac05187)
- Estate Manager - (8f5db12ecacf)
- Estate Manager - (b4e4d0f5dff7)
- Estate Manager - (f1450f9ee088)
- Estate Manager Enactor Payment Device Management Service
- Estate Manager Enactor Web Core
- ✓ Estate Manager Enactor Web Maintenance
- Estate Manager Enactor Web Message Service
- Estate Manager Enactor Web Reports
- Estate Manager Enactor Web Retail Processing
- Estate Manager Enactor Web Terminal Management
- Estate Manager Rest API - Web Rest Api
- Order Manager - (58c65e6816ef)
- Order Manager - (9f02b9f9e63b)
- Order Manager Enactor Web Core
- Order Manager Enactor Web Message Service
- Order Manager Enactor Web Order Processing
- Order Manager Rest API - Web Rest Api

After a log is selected, it can be viewed in the Application Log UI.



In cases where an error and stack trace has been generated, a binocular icon is displayed next to the error – selecting this allows the user to view the stack trace in full.



It is possible to download and view the 'raw' log by selecting the **Save Log File** button, then opening the file in a log viewer or text editor.

The Estate Manager is structured in a very similar way to the Store Server as described earlier. The logs described below are most likely to be of interest in diagnosing issues.

### [Estate Manager Enactor Web Core](#)

The Estate Manager Enactor Web Core log is generated by the Enactor Web Core application and can be used for troubleshooting any issues with the Enactor core services.

### [Estate Manager Enactor Web Reports](#)

The Estate Manager Enactor Web Reports log is generated by the Enactor Web Reports application and can be used for troubleshooting report issues.

### [Estate Manager Enactor Web Maintenance](#)

The Estate Manager Enactor Web maintenance log is generated by the Enactor Web Maintenance application. This log can be used for diagnosing issues with the numerous applications contained within the Estate Manager web UI.

### [Estate Manager Enactor Web Retail Processing](#)

The Estate Manager Enactor Web Retail Processing log is generated by the Enactor Web Retail Processing application. This log can be used for diagnosing issues with background processing (e.g., sending broadcasts or document processing).

### [Card Authorisation Server Enactor Web Card Authorisation](#)

The Card Authorisation Server Enactor Web Card Authorisation log is generated by the Enactor Web Card Authorisation application. This log can primarily be used for diagnosing issues with the Enactor gift card service.

## Application Process Log

The Application Process Log records Events deemed significant at the level of Process design. Application Processes are often executed as scheduled jobs. However, Application Processes running as background services often consume message entries from one or more queues, processes the objects received and post outputs to another queue or queues. Although Application Process Log entries are generated sequentially, they are ordered for presentation according to the Process Execution in which they occur. The Application Process Log is available on the Estate Manager and on each Store Server device.

The Application Process Log page shows details of application process activity. As illustrated in the example below, each entry is listed in the log, grouped for the Process showing Type (also using the Error (❌), Warning (⚠️) or Informational (i) icons, Date (and time) of the message, the Message, Device ID, Entry Type, Reference ID, Process ID and User ID.

The screenshot shows the 'Application Process Log' interface. It features a top navigation bar with a title and window controls. Below the navigation bar is a filter section with various dropdown menus and input fields for filtering log entries based on Start Date, End Date, Log Entry Type, Region, Location, User ID, Device, Message Base, Message ID, Message Text, Entry Type, Reference ID, Device ID, Process ID, Exception Summary, and Exception Detail. There are also 'Apply Filters' and 'Reset Filters' buttons. The main area contains a table of log entries with columns for icons, date/time, message text, actor, action, reference ID, process ID, and process name. The table shows various events such as 'The Card Authorisation Service has Failed', 'Job Started', 'Job Succeeded', and 'Message with id 273,885 from ServerJobResultsIn queue failed to process...'. At the bottom, there are navigation controls for page 2 of 230, page size 999, and an 'Export Results' button.

The following filters are available in the Application Process Log:

Filter	Description
<b>Log Entry Type</b>	Dropdown selection from a list - <b>Errors, Warnings, Information or Debug.</b>
<b>Start Date</b>	Beginning of a Date Range selection within which entries occurred
<b>End Date</b>	End of a Date Range selection within which entries occurred.
<b>Region</b>	Dropdown selection from a list of configured Regions.
<b>Location</b>	Dropdown selection from a list of configured Locations.
<b>User ID</b>	Select entries based on the User Account under which the Application Process was executing.
<b>Device</b>	Dropdown selection from a list of configured Devices. Selects entries for the Device on which they occurred
<b>Device ID</b>	Allows filtering based on a partial or full device ID.
<b>Entry Type</b>	The way the Application Process has been executed (usually <i>scheduledJob</i> ).

<b>Reference ID</b>	This refers to the ID of the scheduled job that triggered the log entry.
<b>Process ID</b>	The ID of the Application Process executing (e.g., DayStartJob).

### Application Process Log - View

Individual entries can be viewed using the View icon (👁) in the left column of the list. The log information is presented in tabs available in the View page depending on the type of Log Entry. All fields are read-only.

### Application Process Log -General Tab View

The General Tab shows general details of the selected application process log entry. The fields are:

Property	Description
<b>Type</b>	The type of log entry e.g. Error.
<b>Date</b>	Date and time at which the log entry occurred.
<b>Entry Type</b>	Type of activity which caused the log entry e.g. businessProcess.
<b>Reference ID</b>	ID of the reference for the log entry.
<b>Process ID</b>	ID of the process which caused the log entry e.g. Task/ServerProcessRunner.
<b>Device ID</b>	ID of the device where the problem occurred.
<b>User ID</b>	ID of the User associated with the problem.
<b>Message ID</b>	ID of the message in the file specified in the following field e.g. GENERATE_ADJUSTMENT_PDF_ERROR_2.
<b>Message Base</b>	The file which contains the message detailed in the previous field e.g. InventoryManagement/Adjustments/AdjustStockAtLocationMessages.
<b>Message</b>	The message generated using the message ID and message base defined in the previous fields.

### Application Process Log -Exception Summary Tab View

The Exception Summary Tab shows summary details of the exception that occurred corresponding to the selected log entry.



### Application Process Log -Exception Details Tab View

The Exception Details Tab shows exception details of the selected application process log entry.

[← Back](#)

You are viewing log entry: '1046-58c67c1b81-6e88a043:51950519969e99c4'

General	Exception Summary	Exception Details
<p>Exception Detail</p> <pre>com.enactor.core.ui.processes.LocalizedUIProcessException: Failed to save the data. ErrorCode:'AlreadyExists' (Caused by com.enactor.core.database.RecordAlreadyExistsException: An item with that key already exists. Entity='deviceStatus' Key='deviceId=estateManager.enactor:calendarDate=11 Oct 2023;' in Table='DeviceStatuses' ErrorCode:'AlreadyExists' (Caused by java.sql.SQLIntegrityConstraintViolationException: Duplicate entry 'estateManager.enactor-2023-10-11 00:00:00' for key 'DeviceStatusesKeyIndex') (vendorErrorCode: 1062) (problemStatement: [saveStmt='com.enactor.core.database.PreparedStatementProxy@14155e3c;joinTableStmt='null']]) at com.enactor.common.ui.entities.processes.SaveEntityAction.execute(SaveEntityAction.java:127) at com.enactor.core.ui.actions.UIActionExecutor.execute(UIActionExecutor.java:119) at com.enactor.core.ui.states.UISaveState.executeAction(UISaveState.java:788) at com.enactor.core.ui.states.UISaveState.executeAction(UISaveState.java:691) at com.enactor.core.ui.processes.UIProcess.executeAction(UIProcess.java:1390) at com.enactor.core.ui.processes.UIProcessRunner.executeProcessLink(UIProcessRunner.java:464) at com.enactor.core.ui.processes.UIProcess.call(UIProcess.java:2412) at com.enactor.core.ui.processes.UIProcess.start(UIProcess.java:332) at com.enactor.core.ui.processes.UIProcessRunner.doStartProcess(UIProcessRunner.java:228) at com.enactor.core.ui.processes.UIProcessRunner.startProcess(UIProcessRunner.java:215) at com.enactor.core.ui.processes.UIProcessRunner.startProcess(UIProcessRunner.java:180) at com.enactor.core.ui.actions.BackgroundProcessRunner.run(BackgroundProcessRunner.java:84) Caused by: com.enactor.core.database.RecordAlreadyExistsException: An item with that key already exists. Entity='deviceStatus' Key='deviceId=estateManager.enactor:calendarDate=11 Oct 2023;' in Table='DeviceStatuses' ErrorCode:'AlreadyExists' (Caused by java.sql.SQLIntegrityConstraintViolationException: Duplicate entry 'estateManager.enactor-2023-10-11 00:00:00' for key 'DeviceStatusesKeyIndex') (vendorErrorCode: 1062) (problemStatement: [saveStmt='com.enactor.core.database.PreparedStatementProxy@14155e3c;joinTableStmt='null']]) at com.enactor.core.database.queryTranslators.MySQLQueryTranslator.translateException(MySQLQueryTranslator.java:80) at com.enactor.core.persistence.DatabaseEntityPersister.save(DatabaseEntityPersister.java:362) at com.enactor.core.database.SimpleDataAccessSession.save(SimpleDataAccessSession.java:229) at com.enactor.core.servers.BaseDBServerImpl.doSave(BaseDBServerImpl.java:892) at com.enactor.core.servers.BaseDBServerImpl.save(BaseDBServerImpl.java:866) at com.enactor.core.servers.BaseDBServer.save(BaseDBServer.java:155) at com.enactor.common.ui.entities.processes.SaveEntityAction.saveEntity(SaveEntityAction.java:189) at com.enactor.common.ui.entities.processes.SaveEntityAction.saveEntity(SaveEntityAction.java:158) at com.enactor.common.ui.entities.processes.SaveEntityAction.execute(SaveEntityAction.java:121) ... 12 more Caused by: java.sql.SQLIntegrityConstraintViolationException: Duplicate entry 'estateManager.enactor-2023-10-11 00:00:00' for key 'DeviceStatusesKeyIndex'</pre>		

## Changing Logging Configuration

It is possible to change logging levels and define specific logging on all devices except the Estate Manager.

If an issue is detected, Enactor may ask for some specific class or process specific logging to be switched on. If this is the case, Enactor will specify which log properties file that this should be added to.

Note that once the changes to the log properties file have been made, the relevant application should be restarted to pick up the change.

### Class Logging

Add an entry for the class in the form `{class}.level=LOG_{level}` where `{class}` is the class name and `{level}` is the log level. The below example logs **debug** for all **com.enactor** classes. Amend `com.enactor.level` to the specific class to be debugged.

```
com.enactor.level=LOG_DEBUG
```

### Useful Class Logging Examples

#### Promotion Engine Code Debug Logging

```
# Following turns on debug logging of the Promotion Engine at code level.
com.enactor.mfc.promotion.engine.PromotionEngine.level=LOG_DEBUG
com.enactor.mfc.promotion.engine.PromotionResponseTree.level=LOG_DEBUG
com.enactor.mfc.promotion.engine.PromotionCombinations.level=LOG_DEBUG
```

#### Database performance logging

Writes out performance logging for the database in the Performance/SQL directory

```
com.enactor.core.management.PerformanceLogger.level=LOG_DEBUG
com.enactor.core.database.PerformanceLogger.level=LOG_DEBUG
```

#### Privilege logging

Enables writing out to the log the missing privilege when an operator doesn't have permission to perform a function.

```
com.enactor.coreUI.actions.CheckPrivilegesAction.level=LOG_DEBUG
com.enactor.coreUI.actions.CheckPrivilegesForUserAction.level=LOG_DEBUG
```

#### Peripheral logging

Enables debug logging for any peripherals attached to the POS.

```
com.enactor.peripherals.level=LOG_DEBUG
```

## Process Logging

The following entries need to be added to the log properties file to enable process logging, where `Process.level` can be changed to `Process.<parent process to log>.level` to monitor a specific process, e.g. `Process.Pos.SignOn.SignOn.level=LOG_DEBUG`. Note that the process id uses dots instead of slashes.

```
#Turn on process debugging (the following 3 coreUI classes must be set to LOG_DEBUG
for it to be activated)
com.enactor.coreUI.processes.UIProcess.level=LOG_DEBUG
com.enactor.coreUI.states.UIState.level=LOG_DEBUG
com.enactor.coreUI.actions.UIActionExecutor.level=LOG_DEBUG
Process.<parent process to log>.level=LOG_DEBUG
```

If this is required, Enactor will provide name of the process(es) to be added to the relevant log configuration file.

Again, once the changes to the log properties file have been made, the relevant application should be restarted to pick up the change.



# About This Document

©2023 Enactor Ltd

All documents produced by Enactor Ltd are supplied to customers subject to Copyright, commercial confidentiality and contractual agreements and must not be copied or transferred without permission.

The amendment history of this document can be found in the table below.

## Current Document Version information

Document Version 1.1

Product Version 2.7

## Document Context

This document is part of the Enactor Product Documentation Series. All Enactor products include a comprehensive documentation set designed to improve understanding of the product and facilitate ease of use.

## Document Inquiries

At Enactor we aspire to producing the highest quality documentation to reflect and enhance the quality of our product. If you find that the document is inaccurate or deficient in any way, please assist us in improving our standard by letting us know.

For matters of document quality or any other inquiries regarding this document please contact:

By Email: [documentation@enactor.co.uk](mailto:documentation@enactor.co.uk)

# Document History

The following versions of the document have been produced:

VERSION	STATUS	ISSUE DATE	AUTHOR	REASON FOR ISSUE
1.0	Initial Draft	01/09/24		Initial version
1.1		10/09/24	John Wood	Added detail around Application Process Log