# Oracle® Retail EFTLink

Framework Installation and Configuration Guide
Release 15.0

E69278-02

May 2019

ORACLE

Oracle® Retail EFTLink Framework Installation and Configuration Guide, Release 15.0



Primary Author: Tracy Gunston

Contributors: Matt Preston, Ian Williams

## Value-Added Reseller (VAR) Language

### Oracle Retail VAR Applications

The following restrictions and provisions only apply to the programs referred to in this section and licensed to you. You acknowledge that the programs may contain third party software (VAR applications) licensed to Oracle. Depending upon your product and its version number, the VAR applications may include:

(i) the **MicroStrategy** Components developed and licensed by MicroStrategy Services Corporation (MicroStrategy) of McLean, Virginia to Oracle and imbedded in the MicroStrategy for Oracle Retail Data Warehouse and MicroStrategy for Oracle Retail Planning & Optimization applications.

(ii) the **Wavelink** component developed and licensed by Wavelink Corporation (Wavelink) of Kirkland, Washington, to Oracle and imbedded in Oracle Retail Mobile Store Inventory Management.

(iii) the software component known as **Access Via™** licensed by Access Via of Seattle, Washington, and imbedded in Oracle Retail Signs and Oracle Retail Labels and Tags.

(iv) the software component known as **Adobe Flex™** licensed by Adobe Systems Incorporated of San Jose, California, and imbedded in Oracle Retail Promotion Planning & Optimization application.

You acknowledge and confirm that Oracle grants you use of only the object code of the VAR Applications. Oracle will not deliver source code to the VAR Applications to you. Notwithstanding any other term or condition of the agreement and this ordering document, you shall not cause or permit alteration of any VAR Applications. For purposes of this section, "alteration" refers to all alterations, translations, upgrades, enhancements, customizations or modifications of all or any portion of the VAR Applications including all reconfigurations, reassembly or reverse assembly, re-engineering or reverse engineering and recompilations or reverse compilations of the VAR Applications or any derivatives of the VAR Applications. You acknowledge that it shall be a breach of the agreement to utilize the relationship, and/or confidential information of the VAR Applications for purposes of competitive discovery.

The VAR Applications contain trade secrets of Oracle and Oracle's licensors and Customer shall not attempt, cause, or permit the alteration, decompilation, reverse engineering, disassembly or other reduction of the VAR Applications to a human perceivable form. Oracle reserves the right to replace, with functional equivalent software, any of the VAR Applications in future releases of the applicable program.

# Contents

# Send Us Your Comments

Oracle Retail EFTLink, Framework Installation and Configuration Guide, Release 15.0

Oracle welcomes customers' comments and suggestions on the quality and usefulness of this document.

Your feedback is important, and helps us to best meet your needs as a user of our products. For example:

- Are the implementation steps correct and complete?
- Did you understand the context of the procedures?
- Did you find any errors in the information?
- Does the structure of the information help you with your tasks?
- Do you need different information or graphics? If so, where, and in what format?
- Are the examples correct? Do you need more examples?

If you find any errors or have any other suggestions for improvement, then please tell us your name, the name of the company who has licensed our products, the title and part number of the documentation and the chapter, section, and page number (if available).

> **Note:** Before sending us your comments, you might like to check that you have the latest version of the document and if any concerns are already addressed. To do this, access the Online Documentation available on the Oracle Technology Network Web site. It contains the most current Documentation Library plus all documents revised or released recently.

Send your comments to us using the electronic mail address: retail-doc_us@oracle.com

Please give your name, address, electronic mail address, and telephone number (optional).

If you need assistance with Oracle software, then please contact your support representative or Oracle Support Services.

If you require training or instruction in using Oracle software, then please contact your Oracle local office and inquire about our Oracle University offerings. A list of Oracle offices is available on our Web site at www.oracle.com.

# Preface

The *Oracle Retail EFTLink Framework Installation and Configuration Guide* describes the requirements and procedures to install this Oracle Retail EFTLink release.

## Audience

This Installation Guide is for the following audiences:

- System administrators and operations personnel
- Database administrators
- System analysts and programmers
- Integrators and implementation staff personnel

## 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

For more information, see the following documents in the Oracle Retail EFTLink Release 15.0 documentation set:

- *Oracle Retail EFTLink Release Notes*
- *Oracle Retail EFTLink Core Configuration Guide*
- *Oracle Retail EFTLink Security Guide*

## Customer Support

To contact Oracle Customer Support, access My Oracle Support at the following URL:

https://support.oracle.com

When contacting Customer Support, please provide the following:

- Product version and program/module name
- Functional and technical description of the problem (include business impact)
- Detailed step-by-step instructions to re-create
- Exact error message received
- Screen shots of each step you take

## Review Patch Documentation

When you install the application for the first time, you install either a base release (for example, 15.0) or a later patch release (for example, 15.1). If you are installing the base release or additional patch releases, read the documentation for all releases that have occurred since the base release before you begin installation. Documentation for patch releases can contain critical information related to the base release, as well as information about code changes since the base release.

## Improved Process for Oracle Retail Documentation Corrections

To more quickly address critical corrections to Oracle Retail documentation content, Oracle Retail documentation may be republished whenever a critical correction is needed. For critical corrections, the republication of an Oracle Retail document may at times **not** be attached to a numbered software release; instead, the Oracle Retail document will simply be replaced on the Oracle Technology Network Web site, or, in the case of Data Models, to the applicable My Oracle Support Documentation container where they reside.

This process will prevent delays in making critical corrections available to customers. For the customer, it means that before you begin installation, you must verify that you have the most recent version of the Oracle Retail documentation set. Oracle Retail documentation is available on the Oracle Technology Network at the following URL:

http://www.oracle.com/technetwork/documentation/oracle-retail-100266.html

An updated version of the applicable Oracle Retail document is indicated by Oracle part number, as well as print date (month and year). An updated version uses the same part number, with a higher-numbered suffix. For example, part number E123456-**02** is an updated version of a document with part number E123456-**01**.

If a more recent version of a document is available, that version supersedes all previous versions.

## Oracle Retail Documentation on the Oracle Technology Network

Oracle Retail product documentation is available on the following web site:

http://www.oracle.com/technetwork/documentation/oracle-retail-100266.html

(Data Model documents are not available through Oracle Technology Network. You can obtain them through My Oracle Support.)

## Conventions

**Navigate:** This is a navigate statement. It tells you how to get to the start of the procedure and ends with a screen shot of the starting point and the statement "the Window Name window opens."

```
This is a code sample
    It is used to display examples of code
```

# 1

# Overview

## Installation Guide Overview

Installation of EFTLink consists of the following steps:

1. Extract the EFTLink files from a zip – `eftlink_v15.0.zip` to a folder on your system.
2. Select one specific core to connect to the EFT system or terminal to be used. Separate batch and script files are provided to do this for each core from a command line for both Windows and Linux.
3. Install EFTLink as a service – a batch file is provided for Windows. For Linux either the EFTLink application can be called at startup or set up as a daemon.
4. Configure the specific core.

This *EFTLink Framework Installation and Configuration Guide* covers the installation and configuration of the framework for EFTLink. Its companion volume, the *EFTLink Core Configuration Guide*, details the specific settings required to configure each Core to communicate with a specific payment system.

## Product Overview

There are multiple manufacturers of Point of Sale (POS) terminals on the market. There are also large numbers of manufacturers of card readers and PIN Entry Devices (PEDs). These card readers can accept a wide variety of cards including debit cards, credit cards, loyalty cards and fuel cards for motor vehicles. These cards are provided by a wide range of issuing organizations each with their own Electronic Payment Systems (EPS). Interconnecting the POS systems, card readers and EPSs is a complex task.

EFTLink is an efficient, platform independent way of providing the connection. It is written in Java, distributed as a Java library and readily added to the software of individual POS terminals.

EFTLink is a router and protocol converter that presents a standard interface to a payment client (typically for a POS) and also links to any card readers or authorization systems in use at the retailer. The interface with the authorization system is therefore separate from the POS, removing any impact of country-specific or server-specific requirements from the POS itself.

EFTLink comes in two parts:

- The EFTLink Framework.
- EFTLink Cores.

The EFTLink Framework provides a system-independent execution environment (a framework) for a targeted EFT solution. The EFTLink Core for a specific terminal or payment system is implemented as a plug-in module that runs within that framework.

Oracle can provide cores for many of the most commonly used card readers or PEDs. Cores can also readily be written for any other card readers or PEDs that require them. Once a core is available for a specific device it will normally work on a range of POSs without further modification.

The POS/EFTLink interface conforms to the Open Payment Initiative (OPI). This is an open standard, widely used in the retail industry. Over time, the original OPI specification has been adopted, extended and maintained by the International Forecourt Standards Forum (IFSF). This enhanced IFSF POS-EPS version is now taken as the definitive specification.

EFTLink is not a full implementation of the IFSF POS-EPS specification. Instead, it uses those parts of the base specification that are pertinent to the sales of dry goods in the retail sector and to the sale of wet goods in petrol (gas) stations. EFTLink includes all the main messages from the IFSF POS-EPS specification and those messages contain all mandatory elements and attributes. EFTLink also includes optional elements and attributes that are commonly used by retailers.

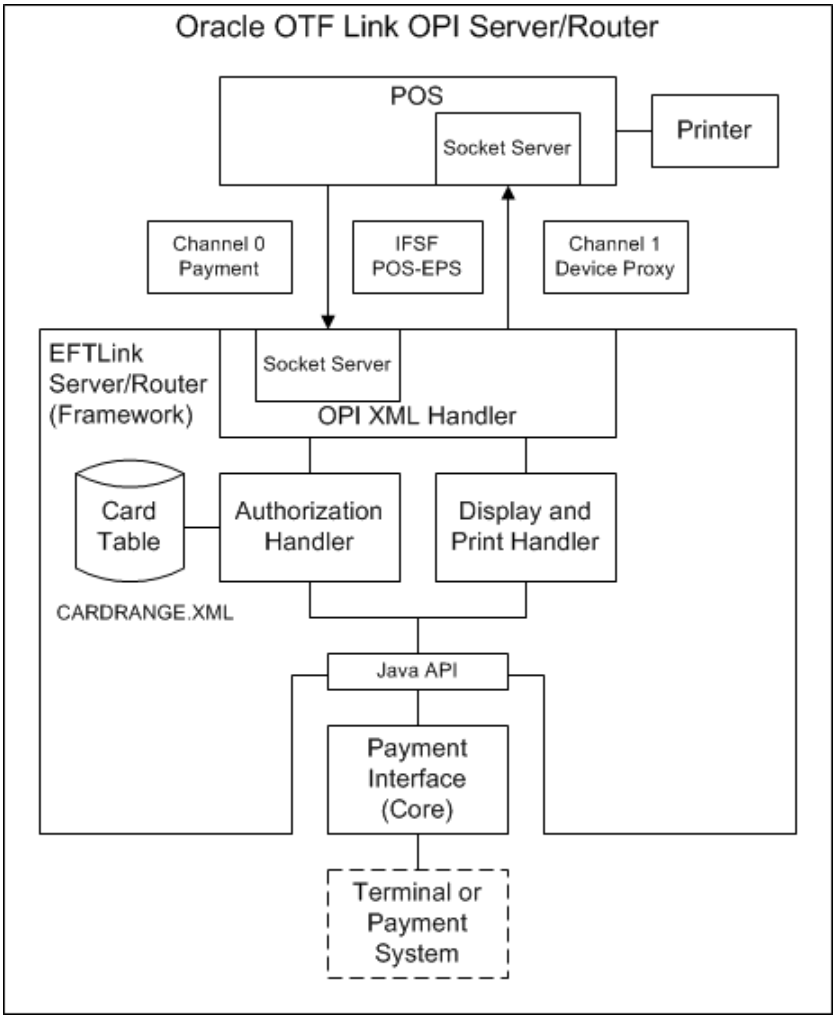
EFTLink can also be extended beyond the IFSF POS-EPS specification. This allows additional features to be included to deal with extended payment or loyalty requirements being driven by new initiatives in retail. This gives considerable flexibility in dealing with the evolving requirements of the future.

Examples of where EFTLink is used include:

- Payment, Refund, Reversal, Pre-authorization and Completion.
- Loyalty Award and Redemption, Balance enquiry, Discount voucher/coupon, IOUs.
- Stored Value Cards – Load, Redeem, Balance enquiry, Activate and so on.
- Online Agents – E-top-up and utilities payments.
- Tokenization, Gratuity, Cashback, DCC, Ad-hoc card read.
- Combined Payment & POS receipts.
- Maintenance functions.
- EPS / PED pooling.

## Architectural Overview

EFTLink is a router and protocol converter, presenting an IFSF/OPI interface to a payment client (typically a POS), and linking to whatever authorization system (or systems) the customer uses. The adoption of a standard IFSF/OPI interface makes EFTLink portable to other POS or payment environments. EFTLink is not in itself a complete solution. What it provides is a system-independent execution environment (a framework) for a targeted EFT solution. The core implementation for a specific terminal or payment system is implemented as a plug-in module that runs within that framework.


Oracle OTF Link OPI Server/Router
POS
Socket Server
Printer
Channel 0
Payment
IFSF
POS-EPS
Channel 1
Device Proxy
EFTLink
Server/Router
(Framework)
Socket Server
OPI XML Handler
Card
Table
Authorization
Handler
Display and
Print Handler
CARDRANGE.XML
Java API
Payment
Interface
(Core)
Terminal or
Payment
System

# 2

# Installation

## Skillset

To install EFTLink successfully implementers must:

- Understand the requirements of the specific EFT system being used, and the POS software that will be connecting to EFTLink.
- The settings that control how EFTLink and the selected core behave are in property files. SIs must know how to add or modify properties within property files with their chosen text editor.
  - Java properties are case sensitive, and never contain spaces in the property name. They usually do not contain spaces in the property value – there are sometimes exceptions in lists.
  - A space is allowed before and after the = that separates the property from its value.
  - Case sensitivity does not apply to Boolean values – True is the same as true.
  - Each property = value is a separate line.
  - Lines prefixed with # are comments.

## Prerequisites

EFTLink can be installed on Windows or Linux operating systems, but the procedure will differ accordingly.

### Java

The EFTLink framework will run with any version of Java from 1.6 on. Individual cores may require a newer version, for example `TransaxEFT` requires 1.7 in this release. To both check the installed java version, and confirm that java is installed so that java.exe can be found by the operating system at a command prompt \ terminal use the command:

```
java –version
```

For both Windows and Linux this returns the full version, provided that java.exe is on the Windows search path, which normally would be the case.

If the command fails, steps will be needed to configure EFTLink to find java.exe, covered below in Installing as a Service.

### POS System Requirements

The POS system should meet the following minimum requirements.

- 256MB RAM
- Intel Celeron 1GHz or equivalent CPU
- 1GB disk space.

### EFTLink Folder

A folder should be created or designated for the EFTLink package. This folder can be any name and location, the only restriction is that there should be no spaces in the path. Conventionally you may wish to use the name `eftlink`.

# Installation Sequence

## Step 1 – Install the Files

EFTLink is supplied as a zip file, `eftlink_v15.0.zip`, and should be unzipped into the designated folder. All files needed, including the entire set of core files are included.

## Fileset

Once unzipped, the following files should be present in the designated EFTLink folder:

- `eftlink.jar`
- `EftLinkConfig.properties`
- `CardRange.xml`
- `installcore.bat`
- `installcore.sh`
- `eftlink.bat`
- `eftlink.sh`
- `cores`
  - `Banksys`
  - `CCVPos`
  - `FIPay`
  - `Ingenico`
  - `OciusSentinel`
  - `PointUS`
  - `SixPay`
  - `SolveConnect`
  - `SteriaPay`
  - `TransaxEFT`
  - `WorldPay`
- `Lib`
  - `jdom-2.0.6.jar`
  - `license.txt`
  - `log4j-1.2.17.jar`
  - `RXTXcomm.jar`
  - `rxtxParallel.dll`
  - `rxtxSerial.dll`
  - `rxtx-source.zip`
- `service-wrapper`
  - `installOPI.bat`
  - `license.txt`
  - `uninstallOPI.bat`
  - `wrapper.conf`
  - `wrapper.dll`
  - `wrapper.exe`
  - `wrapper.jar`

Each core sub-directory contains the core jar file, and reference copies of that core's property file(s).

- The lib folder contains supporting files for EFTLink.
- The service-wrapper folder contains files needed to install EFTLink as a windows service.
- `eftlink.jar` is the main executable code of the EFTLink framework.
- `EftlinkConfig.properties` carries the settings for the framework.
- `CardRange.xml` is the default tender mapping and card identification file.
- `installcore.bat` is a batch file to select one of the cores as active for Windows systems.
- `installcore.sh` is a Linux script to select one of the cores as active for Linux.
- `eftlink.bat` is a batch file to start EFTLink directly (instead of installing as a service) on Windows.
- `eftlink.sh` is a script file to start EFTLink directly on Linux systems.

## Step2 – Select a Core

- For Windows run installcore.bat
- For Linux run installcore.sh

```
usage: installcore [corename]
available cores:
Ingenico        - Ingenico France Telecom
SixPay          - Six Payment Services MPD
SolveConnect    - The Logic Group SolveConnect
OciusSentinel   - Verifone Ocius Sentinel
SteriaPay       - Point (Scandinavia) SteriaPay/PayPoint
PointUS         - Verifone Point (US)
WorldPay        - WorldPay
FIPay           - AJB FIPay
TransaxEFT      - FIS TransaxEFT
CCVPos          - CCVPos ITS
Banksys         - Banksys VIC
```

Example:

```
installcore pointus
```

Would set the PointUS core as the active core.

> **Note**: The core name is not case sensitive in the batch file or Linux script.

The batch or script file does 2 things:

- Configure `EftlinkConfig.properties`:
  `EPSCore0=manito.eft.pointus.PointUSCore`
- Copies the selected core property file from the specific core folder to the main EFTLink folder, where it will be the active file, in this instance `pointus.properties`

  If this is done manually you would need to edit EftLinkConfig.properties

  `EPSCore0=`

  The value is the full classpath to the selected core application. These are the correct classpaths:

| Core | Classpath |
|---|---|
| Six Payment Services MPD | manito.eft.sixpay.SixpayMPDOPIClient |
| The Logic Group SolveConnect | manito.eft.solveconnect.SolveConnectCore |
| Verifone Ocius Sentinel | manito.eft.ocius_sentinel.OciusSentinelCore |
| Point (Scandinavia) SteriaPay/PayPoint | manito.eft.steriapay.SteriaPayCore |
| Verifone Point US | manito.eft.pointus.PointUSCore |
| World Pay | manito.eft.worldpay.WorldPayCore |
| AJB FIPay | manito.eft.ajb.FIPayCore |
| FIS TransaxEFT | manito.eft.transaxeft.TransaxEFTOPIClient |
| CCV Pos ITS | manito.eft.ccvpos.CcvPosCore |
| Banksys VIC | com.torexretail.eftlink.core.vic.VicCore |

## Step 3 – Installing as a Service

### Windows

If the command `java -version` is successful, in the service-wrapper folder run installOPI.bat

This will create a Windows service, OPI Server, to run the EFTLink framework. The configuration of the service is defined in wrapper.conf, and includes setting the classpath for all the supplied cores, in the locations supplied in the zip.

The service will be set to start automatically, but will not be started on install. Since there are settings that need to be made for the chosen core, starting the service can be deferred until that is done, or started now to check that it runs.

Once installed the service can be started and stopped from a command line

```
net start opi
net stop opi
```

Or can be controlled from the Windows Services Control Panel applet ("OPI Server").

If the command `java -version` is not successful, check if the system environment variable JAVA_HOME pointing to the parent folder of the bin folder which includes java.exe exists. If it does not, it can be created.

Then edit `wrapper.conf` with a text editor

### Default

```
# Java Application
#wrapper.java.command=%JAVA_HOME%/bin/java
wrapper.java.command=java
```

### With JAVA_HOME

Swap the comment to make the `JAVA_HOME` option active:

```
# Java Application
wrapper.java.command=%JAVA_HOME%/bin/java
#wrapper.java.command=java
```

### Without JAVA_HOME

If for any reason setting `JAVA_HOME` is undesirable a full path to java.exe can be entered:

```
# Java Application
wrapper.java.command=C:/Program Files/Java/jre1.8.0_60/bin/java
#wrapper.java.command=java
```

If setting the full path, an update to the java version may mean changing `wrapper.conf` Changes to `wrapper.conf` are applied on a service restart.

### Wrapper.log

In the service-wrapper folder installing, starting the service, stopping the service, and uninstalling the service are all briefly logged in wrapper.log, and this can be used to diagnose any problems.

### Linux

Daemon installation is not available at this time.

## Step 4 – Configuring the Core

Go to EFTLink Core Configuration Guide; refer to the section for the core selected.

# 3

# EFTLink Configurable Properties

## Configuration Settings

Settings are defined in `EftlinkConfig.properties`.

### Key Settings

These settings must be set for all POSs

#### EPSCore0

Name of EPS subsystem

Plugin cores must be specified by their full package name, and the package must be added to the execution classpath

Example

```
EPSCore0 = manito.eft.pointus.PointUSCore
```

> **Note**: Although a key setting, EPSCore0 is set by `installcore.bat` / `installcore.sh`

#### DisplayLanguage

Language for display texts. For whatever country code is set, there must be a matching LangXX.properties file. A hierarchy is implied e.g. EN_US is taken as an extension of EN
Example

```
DisplayLanguage = EN
```

### Secondary Settings

These settings are normally correct at their default values, but can be overridden if necessary

#### ServerChannel0

Socket that EFTlink listens on for incoming Channel 0 requests from POS

Example

```
ServerChannel0 = 10100
```

#### ServerChannel1

Socket that EFTlink uses to send Channel 1 Device Requests to POS

Example

```
ServerChannel1 = 10101
```

#### Channel1IP

IP that EFTlink uses to send Channel 1 Device Requests to POS. Default is 127.0.0.1 as EFTLink is normally run on the same system as the POS.

Example

```
Channel1IP = 192.168.0.101
```

### DynamicConfiguration

Static/Dynamic Configuration

Eftlink can be configured to pick up its configuration dynamically from POS messages. A default setting is implied by the POS type setting, but this can be overridden.

Example

```
DynamicConfiguration = false
```

### DecimalPlaces

Number of decimal places to show

Example

```
DecimalPlaces = 2
```

### ShowPrintingDialog

Whether to precede each print request with a TXT_PRINTING (e.g. "Printing. Please Wait") dialog.

Example

```
ShowPrintingDialog = false
```

### DeviceEvents

Whether device events e.g. CardInserted are supported by the POS. Default false.

Example

```
DeviceEvents = false
```

### ValidateItemValues

Whether the basket content should be validated to ensure that the sum of the items matches the overall value. Default true.

Example

```
ValidateItemValues = true
```

### PrinterImpliedOnline

Whether the printer can be assumed to be online and available.
i.e. if the POS can only send requests when the printer is online and with paper, there is no need to do an explicit check.

Example

```
PrinterImpliedOnline = false
```

# 4

# EFTLink General Information

## Tender Mapping

EFTLink provides a table – `CardRange.xml` – for mapping EFT cards to POS tenders. This is done by card IIN range, or, where that is not possible, by card name (also known as card circuit). The resulting numeric code is returned to the POS so that it can determine what tender to allocate the payment to. By default the table maps all card to a single "type" (or tender) by a simple wildcard catchall. This can be used as-is, but if a more detailed breakdown of card type is needed; the relevant card ranges must be added to the file.

CardRange.xml can also be used to map cards by range to a suitable description for display on the receipt. `CardRange.xml` includes comments to explain the layout.

It is anticipated that each POS development team will want to prepare a suitable CardRange.xml for their specific POS requirements, in which case the file can be replaced as required.

## Logging - EFTLink framework & Core

EFTLink uses a standard java logging package - `log4j`. It maintains a daily log file - `EFTLink_mmdd.log` – and deletes log files after 5 days. Both the framework and the core log into this file.

Log files are in the log subdirectory created as soon as EFTLink starts. By default **info** level logging is enabled. This means that key information is logged but the files are kept as small as possible.

To keep files for longer, or increase the logging level, set `log.properties` appropriately. Extract `log.properties` into the main EFTLink directory.

```
unzip eftlink.jar log.properties.
```

For debug logging change the following property.

```
log4j.rootLogger=info, R
```

to

```
log4j.rootLogger=debug, R
```

Logging at debug level does not noticeably affect system performance, but does generate larger log files. To keep logs for longer, edit `log4j.appender.R.MaxBackupIndex=` to a number of days to keep files after the current day (default value is 4)

Consider available disk space when choosing a number of days.

After installing EFTLink as a service, then starting the service, the log file will show about 16 lines, with some basic information, and log that it is deferring all initialization until POS type is known. Once a POS starts, you see details of the core started, with the settings and so on.

# Translation

Most display messages are generated by the core in use or by the host, in which case they are displayed without change. There are also some display messages generated by EFTLink itself. These are defined in `LangEN.properties`, which is embedded in the `EFTLink.jar` file. If necessary, the file can be extracted and edited.

The EFTLink framework supports a number of other languages. Setting EFTLink framework to use one of these is in `EftLinkConfig.properties`

```
DisplayLanguage = EN
```

Possible values are:

| Language | Setting |
|---|---|
| Chinese (Simplified) | CN |
| Dutch | NL |
| English | EN |
| French | FR |
| German | DE |
| Italian | IT |
| Japanese | JP |
| Portuguese | PT |
| Russian | RU |
| Spanish | ES |
| Swedish | SV |

Each of these has its own language property file embedded in eftlink.jar, for example `LangDE.properties`, which if extracted, can be edited as well.

> **Note**: The languages that do not use the Latin alphabet have the characters defined in unicode in their property file. To display messages in Chinese, Japanese or Russian the operating system must support those languages.

Setting `EftlinkConfig.properties DisplayLanguage =` will also control which language a core will use for core specific translations.

| Core | Languages included |
|---|---|
| Banksys VIC | French<br>Dutch<br>English |
| CCVPos ITS | French<br>Dutch<br>English |
| AJB FiPay | English<br>French |
| Verifone Ocius Sentinel | No translations included |

| Core | Languages included |
|---|---|
| Verifone Point US | English |
| SixPayment Services MPD | German<br>English<br>French<br>Italian<br>Dutch |
| The Logic Group SolveConnect | English<br>Spanish |
| Point (Scandinavia) SteriaPay | No translations included |
| FIS TransaxEFT | English |
| World Pay | No translations included |

# A

# Glossary

### Tender

A description or grouping of a payment type. Sometimes called a MOP (Method of Payment)

### Card IIN

The first few numbers of a card PAN that will identify the card type

### Card Circuit

A textual description of the card returned by the payment system, often where the payment system does not return a card IIN

### DCC

Dynamic Currency Conversion. Converting a sale into the home currency of the card holder by the EFT payment system

# B

# Appendix: Installation Order

This section provides a guideline as to the order in which the Oracle Retail applications should be installed. If a retailer has chosen to use some, but not all, of the applications the order is still valid less the applications not being installed.

> **Note:** The installation order is not meant to imply integration between products.

## Enterprise Installation Order

1. Oracle Retail Merchandising System (RMS), Oracle Retail Trade Management (RTM)
2. Oracle Retail Sales Audit (ReSA)
3. Oracle Retail Extract, Transform, Load (RETL)
4. Oracle Retail Active Retail Intelligence (ARI)
5. Oracle Retail Warehouse Management System (RWMS)
6. Oracle Retail Invoice Matching (ReIM)
7. Oracle Retail Price Management (RPM)
8. Oracle Retail Allocation
9. Oracle Retail Mobile Merchandising (ORMM)
10. Oracle Retail Xstore Office
11. Oracle Retail Xstore Point-of-Service, including Xstore Point-of-Service for Grocery, and including Xstore Mobile
12. Oracle Retail Xstore Environment
13. Oracle Retail EFTLink
14. Oracle Retail Store Inventory Management (SIM), including Mobile SIM
15. Oracle Retail Predictive Application Server (RPAS)
16. Oracle Retail Batch Script Architecture (BSA)
17. Oracle Retail Demand Forecasting (RDF)
18. Oracle Retail Category Management Planning and Optimization/Macro Space Optimization (CMPO/MSO)
19. Oracle Retail Replenishment Optimization (RO)
20. Oracle Retail Analytic Parameter Calculator Replenishment Optimization (APC RO)
21. Oracle Retail Regular Price Optimization (RPO)
22. Oracle Retail Merchandise Financial Planning (MFP)
23. Oracle Retail Size Profile Optimization (SPO)
24. Oracle Retail Assortment Planning (AP)
25. Oracle Retail Item Planning (IP)
26. Oracle Retail Item Planning Configured for COE (IP COE)
27. Oracle Retail Advanced Inventory Planning (AIP)
28. Oracle Retail Integration Bus (RIB)
29. Oracle Retail Services Backbone (RSB)
30. Oracle Retail Financial Integration (ORFI)
31. Oracle Retail Data Extractor for Merchandising

32. Oracle Retail Clearance Optimization Engine (COE)
33. Oracle Retail Analytic Parameter Calculator for Regular Price Optimization (APC-RPO)
34. Oracle Retail Insights, including Retail Merchandising Insights (previously Retail Merchandising Analytics) and Retail Customer Insights (previously Retail Customer Analytics)