Oracle® Retail EFTLink

Framework Installation and Configuration Guide
Release 16.0

E78223-07

May 2019

ORACLE

Oracle® Retail EFTLink Framework Installation and Configuration Guide, Release 16.0

E78223-07

Primary Author: Tracy Gunston

Contributors: Matthew 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 16.0

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 16.0 documentation set:

- *Oracle Retail EFTLink Release Notes*
- *Oracle Retail EFTLink Security Guide*
- *Oracle Retail EFTLink Core Configuration Guide*
- *Oracle Retail EFTLink Validated Partner Cores Guide*
- *Oracle Retail EFTLink Configuration Utility User 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, 16.0) or a later patch release (for example, 16.0.1). 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_v16.0.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.

The *Oracle Retail EFTLink Framework Installation Guide* covers the installation and configuration of the framework for EFTLink. A companion volume, the *Oracle Retail 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.

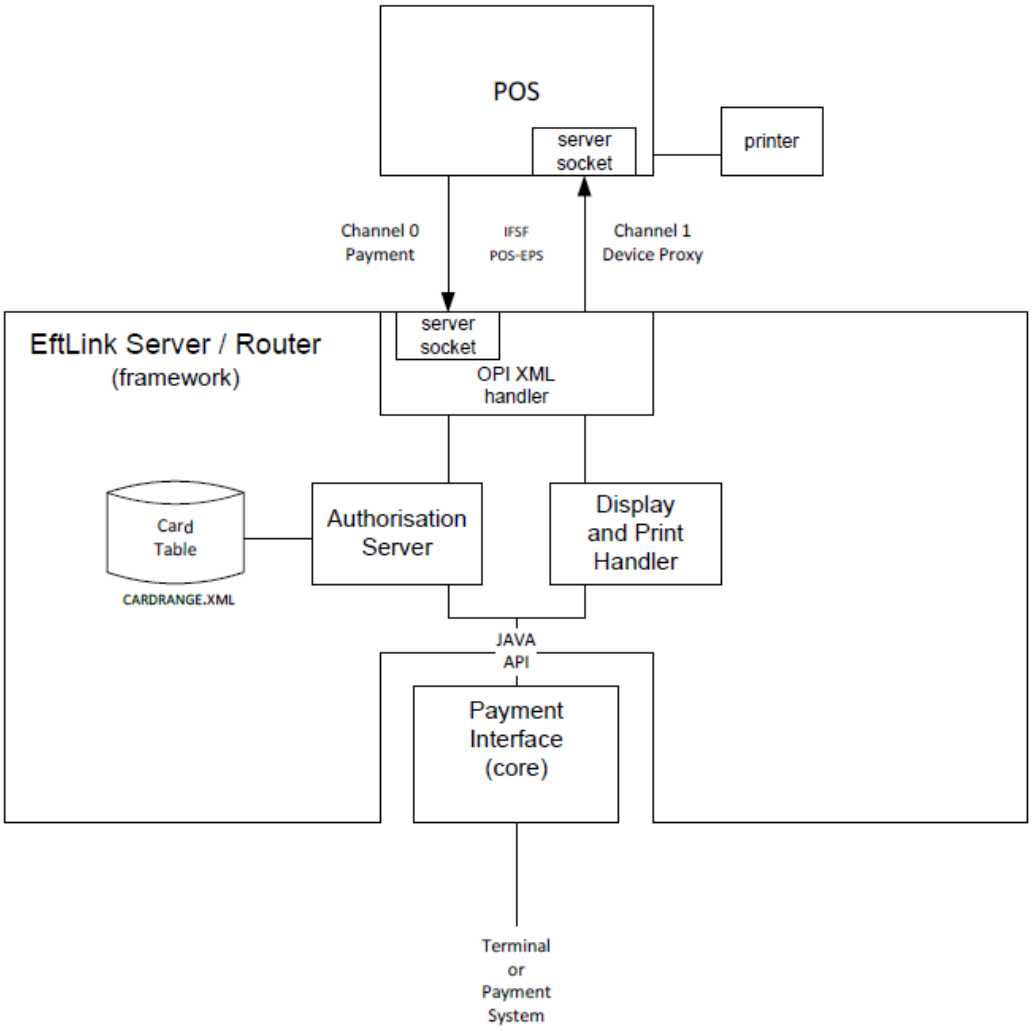
EFTLink can also be extended beyond the IFSF POS-EPS specification. This allows additional features to be included to deal with extended payment or loyalty requirements being driven by new initiatives in retail. This gives considerable flexibility in dealing with the evolving requirements of the future.

Examples of where EFTLink is used include:

- Payment, Refund, Reversal, Pre-authorization and Completion.
- Loyalty Award and Redemption, Balance inquiry, Discount voucher/coupon, IOUs.
- Stored Value Cards – Load, Redeem, Balance inquiry, Activate and so on.
- Online Agents – E-top-up and utilities payments.
- Tokenization, Gratuity, Cashback, DCC, Ad-hoc card read.
- Combined Payment and 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 EFTLink OPI Server/Router

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.

> **Note:** Oracle Retail assumes that the retailer has ensured its Operating System has been patched with all applicable Windows updates.

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_v16.0.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`
- `EFTLinkConfig.bat`
- `CardRange.xml`
- `installcore.bat`
- `installcore.sh`
- `eftlink.bat`
- `eftlink.sh`
- `log4j2.xml`
- `cores`
  - `Adyen`
  - `Banksys`
  - `Cayan`
  - `CCVPos`
  - `FIPay`
  - `Ingenico`
  - `MerchantLink`
  - `OciusSentinel`
  - `PointUS`
  - `SixPay`
  - `SolveConnect`
  - `SteriaPay`
  - `TransaxEFT`
  - `WorldPay`
- `Lib`
  - `jdom-2.0.6.jar`

  - license.txt
  - log4j-1.2-api-2.5.jar
  - log4j-api-2.5.jar
  - log4j-core-2.5.jar
  - RXTXcomm.jar
  - rxtxParallel.dll
  - rxtxSerial.dll
  - rxtx-source.zip
- EFTLinkConfig
  - Application
  - Log
  - Plugin
  - Profile
  - Preprocessor.xml
  - Settings.xml
- Xstore Client
  - Eftlink.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 EFTLinkConfig folder contains the executables and configuration files for the configuration tool.
- The Xstore Client contains a copy of the eftlink.jar for inclusion in the Xstore software.
- `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.
- `EFTLinkConfig.bat` runs the configuration tool which can assist with changes to the configuration.
- `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.

Step 2 – Select a Core

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

```
usage: installcore [corename]
available cores:
Ingenico        - Ingenico
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
Adyen           - Adyen
Cayan           - Cayan
MerchantLink    - Merchant Link
```

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:

- Configures `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
Adyen	`manito.eft.adyen.AdyenCore`
AJB FIPay	`manito.eft.ajb.FIPayCore`
Banksys VIC	`com.torexretail.eftlink.core.vic.VicCore`
Cayan	`manito.eft.cayan.CayanCore`
CCV Pos ITS	`manito.eft.ccvpos.CcvPosCore`
FIS TransaxEFT	`manito.eft.transaxeft.TransaxEFTOPIClient`
Ingenico	`manito.eft.ingenico.IngenicoCore`
Merchant Link	`manito.eft.poslynx.PoslynxCore`
Point (Scandinavia) SteriaPay/PayPoint	`manito.eft.steriapay.SteriaPayCore`
Six Payment Services MPD	`manito.eft.sixpay.SixpayMPDOPIClient`
The Logic Group SolveConnect	`manito.eft.solveconnect.SolveConnectCore`
Verifone Ocius Sentinel	`manito.eft.ocius_sentinel.OciusSentinelCore`
Verifone Point US	`manito.eft.pointus.PointUSCore`
World Pay	`manito.eft.worldpay.WorldPayCore`

Step 3 – Installing as a Service

Windows

It is possible to install EFTLink as a windows service, using a third party wrapper. Previous versions of EFTLink included a version of Tanuki Software Limited Java Service Wrapper, but changes in their licensing terms preclude Oracle from distributing the wrapper.

However there is a version of the wrapper software available to download on the Tanuki website that is free to use under a GPL v2 license.

Download Java Service Wrapper version 3.5.30 For Windows x86 Community Edition from:

http://wrapper.tanukisoftware.com/doc/english/download.jsp

The downloaded file is wrapper-windows-x86-32-3.5.30.zip

Unzip this file into the designated EFTLink folder, so that the designated EFTLink folder has a sub-folder wrapper-windows-x86-32-3.5.30

Configuration for the Java Service Wrapper

Edit the file Designated EFTLinkFolder\ wrapper-windows-x86-32-3.5.30\conf\wrapper.conf with a text editor.

Locate the following property and set:

```
wrapper.java.mainclass=org.tanukisoftware.wrapper.WrapperSimpleApp
```

Update the Java Classpath list as follows:

```
wrapper.java.classpath.1=../lib/wrapper.jar
wrapper.java.classpath.2=../../eftlink.jar
wrapper.java.classpath.3=../../cores/Banksys/banksysvic.jar
wrapper.java.classpath.4=../../cores/CCVPos/ccvposcore.jar
wrapper.java.classpath.5=../../cores/FIPay/FIPayCore.jar
wrapper.java.classpath.6=../../cores/FIPay/AJBComm.jar
wrapper.java.classpath.7=../../cores/Ingenico/ingenicoCore.jar
wrapper.java.classpath.8=../../cores/OciusSentinel/ociussentinelcore.jar
wrapper.java.classpath.9=../../cores/PointUS/pointuscore.jar
wrapper.java.classpath.10=../../cores/SixPay/sixpaycore.jar
wrapper.java.classpath.11=../../cores/SolveConnect/SolveConnect.jar
wrapper.java.classpath.12=../../cores/SteriaPay/steriapaycore.jar
wrapper.java.classpath.13=../../cores/SteriaPay/PayPoint.jar
wrapper.java.classpath.14=../../cores/TransaxEFT/transaxeftcore.jar
wrapper.java.classpath.15=../../cores/WorldPay/worldpaycore.jar
wrapper.java.classpath.16=../../cores/Adyen/AdyenCore.jar
wrapper.java.classpath.17=../../cores/Cayan/cayancore.jar
wrapper.java.classpath.18=../../cores/MerchantLink/poslynxcore.jar
wrapper.java.classpath.19=../../lib/RXTXcomm.jar
```

Locate and update

```
wrapper.java.additional.1=-Deftlink.root=../..
wrapper.app.parameter.1=manito.eft.opi.server.OPIServer
wrapper.console.title=OPI EFT Server
wrapper.name=OPI
wrapper.displayname=OPI Server
wrapper.description=OPI EFT Authorisation Server
```

This wrapper software assumes the java executable is available on the system path. If for any reason there is no environmental variable JAVA_HOME, or you need to use a specific instance of Java wrapper.conf can also be configured to use that JRE by setting a full path to java.exe:

```
wrapper.java.command=C:/Program Files (x86)/Java/jre1.8.0_111/bin/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.

The service wrapper needs 32 bit Java, so if the machine has a default active 64 bit Java, install a 32 bit version as well, and use this option to select the 32 bit version.

Installing the Service

Once wrapper.conf has been configured, you can install the service

Run Designated EFTLinkFolder\ wrapper-windows-x86-32-3.5.30\bin\InstallTestWrapper-NT.bat

This will create a Windows service, OPI Server, to run the EFTLink framework.

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").

Wrapper.log

Designated EFTLinkFolder\ wrapper-windows-x86-32-3.5.30\logs\wrapper.log

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

See the *Oracle Retail EFTLink Core Configuration Guide* and refer to the chapter for the specific core selected.

EFTLink Server

EFTLink is usually deployed as a service application running on each POS and connecting to a single payment device. To support environments where the POS runs as a thin-client application with restricted local device access, or where the hardware has limited processing power or memory, EFTLink can be deployed in Store Server mode. A single EFTLink application runs on a designated server system and all POSs connect to that one server. EFTLink manages the connections to multiple payment terminals and routes payment requests from each POS on to the relevant device.

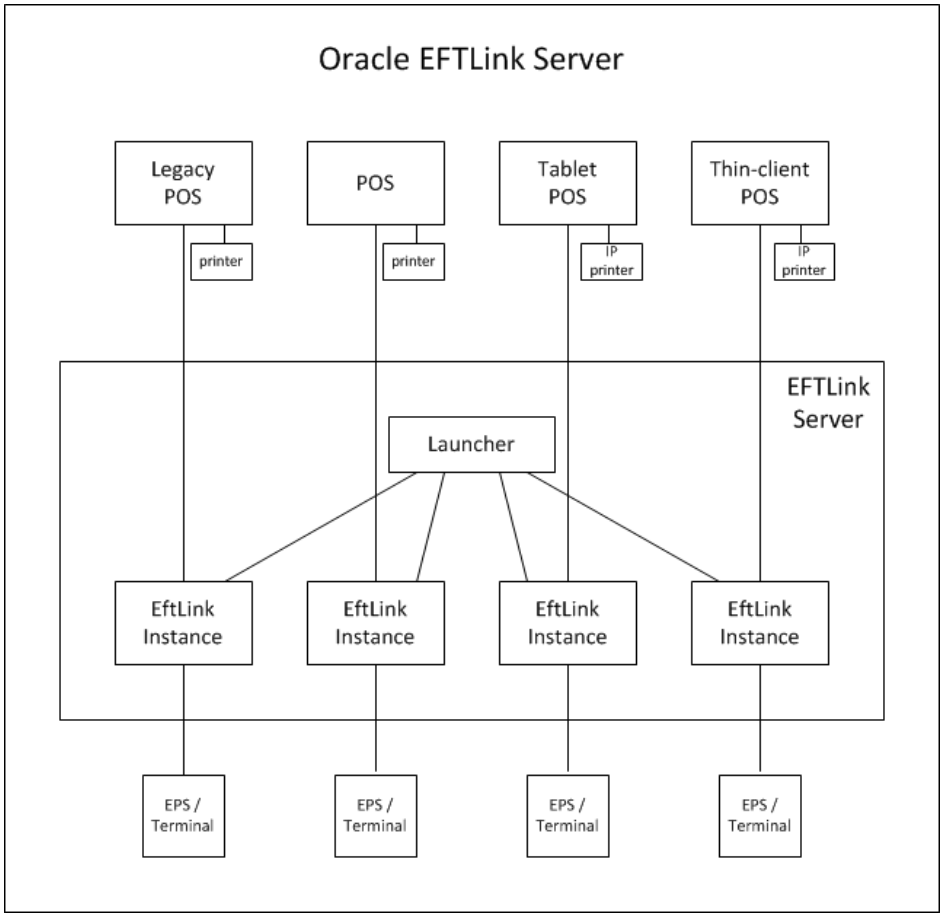
Generally, using Server mode, there is still a 1-1 logical connection between POS and payment terminal, but it is also possible for EFTLink to make a dynamic selection of payment terminal based on availability and convenience. This is referred to as PED-pooling (PED - PIN entry Device).

Similarly, the EFTLink Server can be used to manage a pool of printers shared between the POSs and allocated dynamically. This is referred to as Print-pooling.

This solution is only really possible with IP-based payment terminals and printers. The server system should be in a secure room, and the terminals/printers spread around the store, so direct wired connections are not practical.

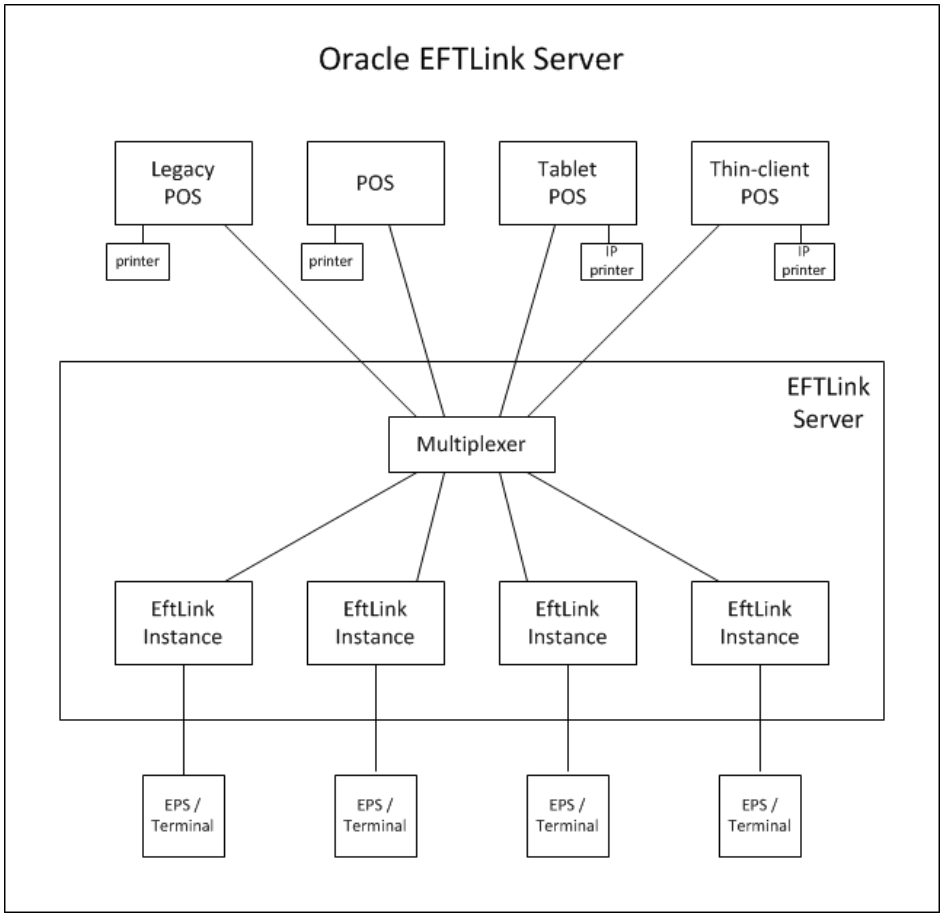
EFTLink Server - Remote Mode

1-1 mapping between the POS and payment system/terminal. Each POS is allocated a fixed pair of sockets (channel 0/1) that connect to a dedicated EFTLink instance.



EFTLink Server - PEDPool Remote Mode

Many-many mapping between POS and payment system/terminal. Each POS is allocated a fixed pair of sockets (channel 0/1) that connect to a multiplexer/switch. The multiplexer implements rules and/or uses interactive dialogs with the POS operator to determine which EFTLink instance to pass the request on to.



Configuring EFTLink Server

Configuring/deploying EFTLink Server is rather more complicated that standard EFTLink and is currently only possible as a manual procedure.

As a base, EFTLink should first be installed on the chosen server system using the standard installation procedure.

Enabling Server Mode

EFTLink Server uses a different main class from normal.

Replace the following lines where applicable:

Windows

Replace: `java manito.eft.opi.server.OPIServer`
With: `java manito.eft.opi.server.MultiServerLauncher`

Linux

Replace: `java -cp $CLASSPATH manito.eft.opi.server.OPIServer`
With: `java -cp $CLASSPATH manito.eft.opi.server.MultiServerLauncher`

Tanuki Wrapper Configuration

Use a text editor to edit service-wrapper/wrapper.conf.

Replace: `wrapper.app.parameter.1=manito.eft.opi.server.OPIServer`
With: `wrapper.app.parameter.1=manito.eft.opi.server.MultiServerLauncher`

PED Pool

Replace: `PEDPoolEnabled = false`
With: `PEDPoolEnabled = true`

See PED Pooling Set Up for more information.

EFTLink Instance Set Up

Each instance of EFTLink is identified by a unique sequence number starting from 1.

For each instance of EFTLink required (that is, for each payment terminal):

1. In the main eftlink folder, run `installcore.bat` as if configuring standalone EFTLink. This will setup the `EftlinkConfig.properties` file.
2. Create a subfolder under the main eftlink folder named serverN, where N is the sequence number.
3. Copy all properties files (*.properties) from the main eftlink folder into the new serverN folder.
4. Using a text editor, edit the core-specific properties file in the subfolder to set any properties that are unique for each core instance for example, the terminal IP.
5. Using a text editor edit `EftlinkConfig.properties` in the main eftlink folder:
    - Find the NumServers setting and change it to be the number of EFTLink instances to be used. Un-comment (that is, remove the leading '# if present) if necessary. For example, NumServers = 2
    - For each EFTLink instance, assign a descriptive title. These are the names that will be presented to the operator and should identify the relevant payment terminal in some way such as by its location, for example:
      server1.description = Menswear-suits
      server2.description = Menswear-paydesk #2 till 1

Note: Spaces are allowed in the descriptive names, but not commas if PED pooling is to be used.

POS Client Set Up

Each POS client is identified by a unique sequence number starting from 1.

1. Use a text editor to edit `EftlinkConfig.properties` in the main eftlink folder:
    a) Find the NumClients setting and change it to be the number of POSs that will be using EFTLink. Un-comment (that is, remove the leading '# if present) if necessary. For example, NumClients = 2
    b) For each POS, assign a descriptive title. These are the names will be shown in the EFTLink log to ease tracking/debugging, for example:
    pos1.description = Menswear-suits
    pos2.description = Menswear-mobile#1
2. Each POS has to use a unique pair of ports for its connection to EFTLink. These do not need to be further defined within `EftlinkConfig.properties`, but the ports numbers and EFTLinkServer system IP must be set on each POS. The numbering system is based on EFTLink base address (default 10100, configurable by the

ServerChannel0 property) plus 10 x the POS number. Two sequential ports are needed, one for each of channel 0 and 1. This gives a default allocation of:

POS1 - 10110/10111
POS2 - 10120/10121
POS3 - 10130/10131
...
POS9 - 10190/10191
POS10 - 10200/10201
POS11 - 10210/10211
and so on

If this range of ports is not available, the base number can be changed via the ServerChannel0 setting. All POSs must then be changed to match.

PED Pooling Set Up

If PED pooling has been enabled, the system uses the standard channel 1 display messages to present each POS operator with a list of available payment terminals. By default, the list will include all available terminals, but this can be confusing in a large store, so there is an option to limit each POS to a subset of the full list to show just the terminals in one department. The subset is defined using the descriptive names from EFTLink Instance Set Up, and specified as a comma-separated list. A default association can be set by prefixing the descriptive name with '*'. If that payment terminal is available, it will be automatically used without any operator prompting.

For example
pos1.subpool = *Menswear-suits
pos2.subpool = Menswear-suits, *Menswear-paydesk #2 till 1, Menswear-paydesk #2 till 2

Note: It is important to point out that the EFTLink PED pooling functionality is restricted by Core compatibility. Please note the following restrictions:

PED pooling is only applicable within the `<CardServiceRequest>` context, that is, this is when the actual payment is initiated and finalized.

PED pooling is not currently applicable within the `<SaleStateNotification>` context, that is, if the EPS supports a device that is dependent on a line display, this functionality will need to be suppressed by Xstore or the Core (depending on configuration).

PED pooling is not possible where the EPS requires the register to be paired with a single device thereby forcing a one to one relationship between the register and the device.

Xstore Set Up

As noted above, each POS has to use a unique pair of ports for its connection to EFTLink. Also, the POS is configured to access a remote EFTLink rather than a local one.

There are two different ways that Xstore can be set up to use with EFTLink in Server Mode.

- One to one port mapping (applies to both Xstore and Xstore Mobile)
- One to many port mapping (applies to Xstore Mobile only)

All configurations illustrated below are part of the Xstore `AuthConfig.xml` configuration file.

One to One Port Mapping

This is where there is one Xstore or Xstore Mobile client served from the Jetty instance. It will divert all requests to a single port pairing that is managed inside the EFTLink Server instance. If another POS client is configured to use the same port pairing, it will potentially be blocked out until the port pair becomes free. In this mode, EFTLink Server will allow a single device to use many PEDs through the PED pooling functionality. EFTLink Server does not support load balancing of requests through one port pair so this configuration is not recommended if there are many Xstore mobile clients in the store solution.

If this configuration is suitable then the Xstore Mobile configuration is identical to the standard Xstore configuration. The 'communicatorHosts' parameter is used to set the channel 0 URL and 'deviceCommChannel' is used to set the channel 1 URL, as illustrated below. In this configuration when Xstore or Xstore Mobile starts an authorization request EFTLink will process the authorization request in the expected way, or if PED pooling is enabled, it will send a list of available PEDs for an associate to choose. Once the associate has chosen a PED, the authorization will proceed in the expected way.

```
<AuthProcess name="EFT_LINK_HOST" Abstract="true">
    <Parameter name="communicatorHosts">
      <param_value dtype="List">
        <Host
dtype="String">socket://localhost:10100;timeout=1000</Host>
      </param_value>
    </Parameter>
    <Parameter name="deviceCommChannel" value="socket://localhost:10101"
/>
...
...
</AuthProcess>
```

One to Many Port Mapping

This configuration is only appropriate for Xstore where XMobile is supported, that is, many XMobile clients are served by the Jetty server. In order to setup Xstore this way an optional parameter called 'additionalWorkstationHostsMap' needs to be configured in addition to the configuration described above. This map will override the default channel 0 and channel 1 values configured in Xstore (see above), where an entry exists for a given workstation id.

```
<AuthProcess name="EFT_LINK_HOST" Abstract="true">
    <Parameter name="communicatorHosts">
      <param_value dtype="List">
        <Host
dtype="String">socket://localhost:10100;timeout=1000</Host>
      </param_value>
    </Parameter>
    <Parameter name="deviceCommChannel" value="socket://localhost:10101"
/>
...
...
    <Parameter name="additionalWorkstationHostsMap">
      <param_value dtype="Map">
        <MapEntry>
```

```
            <key dtype="Integer">1</key> <!-- workstation id -->
            <value dtype="EFTLinkCommunicationChannels">
              <Channel0 dtype="String">socket://localhost:10100</Channel0>
              <Channel1 dtype="String">socket://localhost:10101</Channel1>
            </value>
          </MapEntry>
          <MapEntry>
            <key dtype="Integer">2</key> <!-- workstation id -->
            <value dtype="EFTLinkCommunicationChannels">
              <Channel0 dtype="String">socket://localhost:10200</Channel0>
              <Channel1 dtype="String">socket://localhost:10201</Channel1>
            </value>
          </MapEntry>
        </param_value>
      </Parameter>
    </AuthProcess>
```

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 for example, 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 (for example, "Printing. Please Wait") dialog.

Example

```
ShowPrintingDialog = false
```

DeviceEvents

Whether device events for example, 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, that is, 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
```

MultiJVM

> **Note**: This functionality is currently incompatible with the `PEDPoolEnabled` property.

This property is used to launch each OPIServer in their own Java Virtual Machine (JVM) process when the NumServers property is set to greater than 0.

Each server's channel 0 and channel 1 ports are based on the ServerChannel0 setting. For example; if the ServerChannel0 is set to 10100 and NumServers is set to 3, the additional servers will be created on channel 0 ports 10110, 10120, 10130 and the corresponding channel 1 ports will be 10111, 10121, and 10131 therefore, you must ensure that these ports are available for use with EFTLink.

For each server defined under NumServers; EFTLink looks for a corresponding server folder. For example, if NumServers is set to 3, EFTLink looks for server folders named

server1, server2 and server3 under the EFTLink directory. These folders must contain their own configuration files, that is; `EftLinkConfig.properties` and so on.

In order to use this property you must use the MultiServerLauncher application rather than the OPIServer application.

> **Notes:** It is not recommended to use this functionality with compact systems where memory is at a premium. Therefore, taking the above example of 3 servers running in separate JVMs with each JVM taking roughly ~60MB of memory, EFTLink would require at least ~180MB of free memory (this is a purely hypothetical situation, actual memory usage may be system-dependent).
>
> There may be additional memory requirements dependent upon the core being used with EFTLink.
>
> You must ensure when, using this functionality, that a clean shutdown of EFTLink is performed in order to destroy the child processes which have been created. In Windows command line; the command CTRL+C is used to terminate a batch job cleanly.

Example

```
MultiJVM = 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.

> **Note:** For more information, see the *Oracle Retail EFTLink CardRange.xml Guide* available on My Oracle Support (Doc ID 2266221.1) using the following link:
>
> Oracle Retail EFTLink Supplemental Documentation

Logging - EFTLink Framework and Core

EFTLink uses a standard java logging package - `log4j2`. It maintains a daily log file - `EFTLink_mmdd.log` – and deletes log files after 7 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 `log4j2.xml` appropriately. Extract `log4j2.xml` into the main EFTLink directory.

```
unzip eftlink.jar log4j2.xml
```

For debug logging change the following entry.

```
<Root level="info">
```

to

```
<Root level="debug">
```

Logging at debug level does not noticeably affect system performance, but does generate larger log files. To keep logs for longer, edit

```
<DefaultDateRollOverStrategy max="7"/>
```

to a number of days to keep files after the current day (default value is 7).

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
Adyen	English
AJB FiPay	English French
Banksys VIC	French Dutch English
Cayan	No translations included

Core	Languages included
CCVPos ITS	French Dutch English
FIS TransaxEFT	English
Merchant Link	English
Point (Scandinavia) SteriaPay	No translations included
SixPayment Services MPD	German English French Italian Dutch
The Logic Group SolveConnect	English Spanish
Verifone Ocius Sentinel	No translations included
Verifone Point US	English
World Pay	No translations included

5

Xstore/EFTLink Core Compatibility

This chapter provides information relating to the release versions for compatibility of Xstore and EFTLink for payment terminal functionality in each Core.

International

The following Core is supported:

- Adyen

Adyen

Operating Areas: US, EU, AUS, Singapore

Supported Terminals: Verifone VX 820, e355, MX925

Payment Terminal Function	Subtype	Xstore Release Version	EFTLink Release Version
Sign On		15.0	16.0
Sign Off		15.0	16.0
Card Payment	Payment by Credit	15.0	16.0
	Payment by Debit	15.0	16.0
Refund	Refund by Credit	15.0	16.0
	Refund by Debit	15.0	16.0
	Cardless Refund using Token	16.0	16.0
Reversal	Reversal of last transaction	15.0	16.0
	Reversal of last transaction within batch	15.0	16.0
DCC (currency conversion)	Foreign exchange tender amount, rate, currency code Functionality that is driven through the Core plugin by virtue of the <DeviceRequest> / <DeviceResponse> API.	15.0	16.0
Token Support	Receive and record token for verified return	16.0	16.0
Combined EFT/Tax Receipts	Defer customer copy to <CardServiceResponse>	16.0	16.0
EFTLink server compliant		-	16.0
IP PED		-	16.0

US Region

The following Cores are supported:

- AJB FiPay
- Cayan
- Merchant Link
- Verifone Point US

Where Xstore does not support a function that is supported by one of the Cores, this would imply that a solution may be found in the customer overlay. If functionality is not listed here then assume that it does not exist in the Core EPS API or in the Xstore base.

AJB FiPay

Operating Countries: USA, Canada

Supported Terminals: Verifone MX915, MX925

Note: AJB FiPay has six different SVC providers and therefore each SVC action will need to be checked that the SVC provider supports it.

Payment Terminal Function	Subtype	Xstore Release Version	EFTLink Release Version
Card Payment	Payment by Credit	15.0	15.0
	Payment by Debit	15.0	15.0
Refund	Refund by Credit	15.0	15.0
	Refund by Debit	15.0	15.0
	Cardless Refund using Token	16.0	15.0
Reversal	Reversal of last transaction	15.0	15.0
	Reversal of last transaction within batch	15.0	15.0
Electronic Signature Capture	3 byte ASCII (3BA)	15.0	15.0
Token Support	Receive and record token for verified return	16.0	15.0
Card Swipe – non PCI cards	Get card data	15.0	15.0.1
Combined EFT/Tax Receipts	Defer customer copy to <CardServiceResponse>	16.0	16.0
Customer Display	Sale State Notification	15.0	16.0
IP PED		-	15.0
Stored Value Cards (Gift Cards)	Redeem	16.0	16.0

Payment Terminal Function	Subtype	Xstore Release Version	EFTLink Release Version
	Redeem Reversal	16.0	16.0
	Cashback / Unload	16.0	16.0
	Load	16.0	16.0
	Refund	16.0	16.0
	Cash out (Close)	16.0.1	16.0
	Balance Inquiry	16.0	16.0
	Activate	16.0	16.0
	Activate and Load	16.0	16.0
	Activate Reversal	16.0	16.0
	Deactivate / Stop	-	16.0
	Deactivate / Stop Reversal	-	16.0

Cayan

Operating Countries: USA
Supported Terminals: Verifone MX925

Payment Terminal Function	Subtype	Xstore Release Version	EFTLink Release Version
Sign On		15.0	16.0
Card Payment	Payment by Credit	15.0	16.0
	Payment by Debit	15.0	16.0
Refund	Refund by Credit	15.0	16.0
	Refund by Debit	15.0	16.0
	Cardless Refund using Token	16.0	16.0
Reversal	Reversal of last transaction	15.0	16.0
	Reversal of last transaction within batch	15.0	16.0
Cashback	Cashback prompted on PED, return amount to POS	15.0	16.0
Electronic Signature Capture	3 byte ASCII (3BA)	15.0	16.0
Gratuity		-	16.0
Token Support	Receive and record token for verified return	16.0	16.0
Combined EFT/Tax Receipts	Defer customer copy to <CardServiceResponse>	16.0	16.0

Payment Terminal Function	Subtype	Xstore Release Version	EFTLink Release Version
Customer Display	Sale State Notification	15.0	16.0
IP PED		-	16.0
Stored Value Cards (Gift Cards)	Redeem	16.0	16.0
	Redeem Reversal	16.0	16.0
	Load	16.0	16.0
	Balance Inquiry	16.0	16.0
	Activate Reversal	16.0	16.0

Merchant Link

Operating Countries: USA

Supported Terminals: Verifone MX925

Payment Terminal Function	Subtype	Xstore Release Version	EFTLink Release Version
Card Payment	Payment by Credit	15.0	16.0
	Payment by Debit	15.0	16.0
Refund	Refund by Credit	15.0	16.0
	Refund by Debit	15.0	16.0
	Cardless Refund using Token	16.0	16.0
Reversal	Reversal of last transaction	15.0	16.0
	Reversal of last transaction within batch	15.0	16.0
Reconciliation	Reconciliation with Closure Functionality that is typically available via the hardware maintenance menu on the Xstore POS client, that is, uses the <ServiceRequest> / <ServiceResponse> API.	15.0	16.0
	Manual entry initiated by PED	-	16.0
Electronic Signature Capture	3 byte ASCII (3BA)	15.0	16.0
Token Support	Receive and record token for verified return	16.0	16.0
Card Swipe – non PCI cards	Get card data	15.0	16.0
Combined EFT/Tax Receipts	Defer customer copy to <CardServiceResponse>	16.0	16.0
Customer Display	Sale State Notification	15.0	16.0

Payment Terminal Function	Subtype	Xstore Release Version	EFTLink Release Version
EFTLink server compliant		-	16.0
IP PED		-	16.0
Stored Value Cards (Gift Cards)	Redeem	16.0	16.0
	Redeem Reversal	16.0	16.0
	Cashback / Unload	16.0	16.0
	Load	16.0	16.0
	Refund	16.0	16.0
	Balance Inquiry	16.0	16.0
	Activate	16.0	16.0
	Activate Reversal	16.0	16.0
	Deactivate / Stop	-	16.0
	Deactivate / Stop Reversal	-	16.0

Verifone Point US

Operating Countries: USA

Supported Terminals: Verifone MX915, MX925

Payment Terminal Function	Subtype	Xstore Release Version	EFTLink Release Version
Card Payment	Payment by Credit	15.0	15.0
	Payment by Debit	15.0	15.0
Refund	Refund by Credit	15.0	15.0
	Refund by Debit	15.0	15.0
	Cardless Refund using Token	16.0	15.0.1
	Cardless Refund using Token + Expiry Date	16.0	15.0.1
Reversal	Reversal of last transaction	15.0	15.0
	Reversal of last transaction within batch	15.0	15.0
Reconciliation	Reconciliation with Closure Functionality that is typically available via the hardware maintenance menu on the Xstore POS client, that is, uses the <ServiceRequest> / <ServiceResponse> API.	15.0	15.0
Customer Not Present	CVV2/AVS/partial entry on PED	-	16.0

Payment Terminal Function	Subtype	Xstore Release Version	EFTLink Release Version
Electronic Signature Capture	3 byte ASCII (3BA)	15.0	15.0
Gratuity		-	15.0
Token Support	Receive and record token for verified return	16.0	15.0.1
Card Swipe – non PCI cards	Get card data	15.0	15.0
	Get card data with abort	16.0	15.0.1
Combined EFT/Tax Receipts	Defer customer copy to <CardServiceResponse>	16.0	15.0
Customer Display	Sale State Notification	15.0	15.0
IP PED		-	15.0
Stored Value Cards (Gift Cards)	Redeem	16.0	16.0
	Redeem Reversal	16.0	16.0
	Cashback / Unload	16.0	16.0
	Load	16.0	16.0
	Balance Inquiry	16.0	16.0
	Activate	16.0	16.0
	Activate Reversal	16.0	16.0
	Deactivate / Stop	-	16.0
	Deactivate / Stop Reversal	-	16.0

European Region

The following Cores are supported:

- CCV
- FIS Transax
- Ingenico
- Six Payment Services
- TLG SolveConnect
- Verifone Ocius Sentinel
- Verifone Point NO
- WorldLine (Banksys)
- WorldPay

CCV

Operating Countries: Belgium, Luxembourg, Netherlands
Supported Terminals: Verifone VX 810, VX 820

Payment Terminal Function	Subtype	Xstore Release Version	EFTLink Release Version
Card Payment	Payment by Credit	15.0	15.0
	Payment by Debit	15.0	15.0
Refund	Refund by Credit	15.0	15.0
	Refund by Debit	15.0	15.0
Reversal	Reversal of last transaction	15.0	15.0
	Reversal of last transaction within batch	15.0	15.0
Reconciliation	Reconciliation with Closure Functionality that is typically available via the hardware maintenance menu on the Xstore POS client, that is, uses the <ServiceRequest> / <ServiceResponse> API.	15.0	15.0
Customer Not Present	CVV2/AVS/partial entry on PED	-	15.0
Combined EFT/Tax Receipts	Defer customer copy to <CardServiceResponse>	16.0	15.0
Ticket Reprint (Direct)		-	15.0
IP PED		-	15.0

FIS Transax

Operating Countries: UK
Supported Terminals: Verifone VX 820

Payment Terminal Function	Subtype	Xstore Release Version	EFTLink Release Version
Card Payment	Payment by Credit	15.0	15.0
	Payment by Debit	15.0	15.0
Refund	Refund by Credit	15.0	15.0
	Refund by Debit	15.0	15.0
Reconciliation	Reconciliation with Closure Functionality that is typically available via the hardware maintenance menu on the Xstore POS client, that is, uses the <ServiceRequest> / <ServiceResponse> API.	15.0	15.0
	Create receipt from XML elements provided by EPS	-	15.0
Ticket Reprint (Direct)		-	15.0
Stored Value Cards (Gift Cards)	Balance Inquiry	16.0	15.0

Ingenico

Operating Countries: France, Spain, Portugal
Supported Terminals: Ingenico IPP350

Payment Terminal Function	Subtype	Xstore Release Version	EFTLink Release Version
Sign On		15.0	15.0
Sign Off		15.0	15.0
Card Payment	Payment by Credit	15.0	15.0
	Payment by Debit	15.0	15.0
Check Payment/Check Verification		-	15.0
Refund	Refund by Credit	15.0	15.0
	Refund by Debit	15.0	15.0
Reversal	Reversal of last transaction	15.0	15.0

Payment Terminal Function	Subtype	Xstore Release Version	EFTLink Release Version
Reconciliation	Reconciliation with Closure Functionality that is typically available via the hardware maintenance menu on the Xstore POS client, that is, uses the <ServiceRequest> / <ServiceResponse> API.	15.0	15.0
Card Swipe – non PCI cards	Get card data	15.0	15.0
Ticket Reprint (Direct)		-	15.0
IP PED		-	15.0
Stored Value Cards (Gift Cards)	Redeem	16.0	15.0
	Load	16.0	15.0
	Activate	16.0	15.0
	Activate Reversal	16.0	15.0

Six Payment Services

Operating Countries: Europe

Supported Terminals: Yomani

Payment Terminal Function	Subtype	Xstore Release Version	EFTLink Release Version
Sign On		15.0	15.0
Sign Off		15.0	15.0
Card Payment	Payment by Credit	15.0	15.0
	Payment by Debit	15.0	15.0
Refund	Refund by Credit	15.0	15.0
	Refund by Debit	15.0	15.0
Reversal	Reversal of last transaction	15.0	15.0
	Reversal of last transaction within batch	15.0	15.0
Reconciliation	Reconciliation with Closure Functionality that is typically available via the hardware maintenance menu on the Xstore POS client, that is, uses the <ServiceRequest> / <ServiceResponse> API.	15.0	15.0
Customer Not Present	CVV2/AVS/partial entry on PED	-	15.0

Payment Terminal Function	Subtype	Xstore Release Version	EFTLink Release Version
Combined EFT/Tax Receipts	Defer customer copy to <CardServiceResponse>	16.0	15.0
EFTLink server compliant		-	15.0
IP PED		-	15.0

TLG SolveConnect

Operating Countries: UK

Supported Terminals: Ingenico, Verifone VX 820, VX 680

Payment Terminal Function	Subtype	Xstore Release Version	EFTLink Release Version
Card Payment	Payment by Credit	15.0	15.0
	Payment by Debit	15.0	15.0
Refund	Refund by Credit	15.0	15.0
	Refund by Debit	15.0	15.0
	Cardless Refund using Token	16.0	15.0
Reversal	Reversal of last transaction	15.0	16.0
	Reversal of last transaction within batch	15.0	16.0
DCC (currency conversion)	Foreign exchange tender amount, rate, currency code Functionality that is driven through the Core plugin by virtue of the <DeviceRequest> / <DeviceResponse> API.	15.0	16.0
Cashback	Cashback prompted on PED, return amount to POS	15.0	15.0
Customer Not Present	CVV2/AVS/partial entry on PED	-	15.0
Gratuity		-	15.0
Token Support	Receive and record token for verified return	16.0	15.0
Card Swipe – non PCI cards	Get card data	15.0	15.0
Combined EFT/Tax Receipts	Defer customer copy to <CardServiceResponse>	16.0	16.0
	Suppress Merchant copy for EMV Chip transactions	-	16.0
Ticket Reprint (Direct)		-	15.0

Payment Terminal Function	Subtype	Xstore Release Version	EFTLink Release Version
EFTLink server compliant		-	16.0
IP PED		-	16.0
2 stage payment	Card Inserted Event / update payment	-	15.0
Stored Value Cards (Gift Cards)	Redeem	16.0	15.0
Stored Value Cards (Gift Cards)	Redeem Reversal	16.0	15.0
	Cashback / Unload	16.0	15.0
	Load	16.0	15.0
	Balance Inquiry	16.0	15.0

Verifone Ocius Sentinel

Operating Countries: UK

Supported Terminals: Verifone VX 810, VX 820

Payment Terminal Function	Subtype	Xstore Release Version	EFTLink Release Version
Sign On		15.0	15.0
Sign Off		15.0	15.0
Card Payment	Payment by Credit	15.0	15.0
	Payment by Debit	15.0	15.0
Check Payment/Check Verification		-	15.0
Refund	Refund by Credit	15.0	15.0
	Refund by Debit	15.0	15.0
	Cardless Refund using Token	16.0	15.0
Reconciliation	Reconciliation with Closure Functionality that is typically available via the hardware maintenance menu on the Xstore POS client, that is, uses the <ServiceRequest> / <ServiceResponse> API.	15.0	15.0
DCC (currency conversion)	Foreign exchange tender amount, rate, currency code Functionality that is driven through the Core plugin by virtue of the <DeviceRequest> / <DeviceResponse> API.	15.0	15.0

Payment Terminal Function	Subtype	Xstore Release Version	EFTLink Release Version
Cashback	Cashback prompted on PED, return amount to POS	15.0	15.0
Customer Not Present	CVV2/AVS/partial entry on PED	-	15.0
Gratuity		-	15.0
Token Support	Receive and record token for verified return	16.0	15.0
Card Swipe – non PCI cards	Get card data	15.0	15.0
Combined EFT/Tax Receipts	Defer customer copy to <CardServiceResponse>	16.0	15.0
	Suppress Merchant copy for EMV Chip transactions	-	15.0
	Create receipt from XML elements provided by EPS	-	15.0
Ticket Reprint (Direct)		-	15.0
IP PED		-	15.0
	Card read, card payment	-	15.0
Stored Value Cards (Gift Cards)	Redeem	16.0	15.0
	Redeem Reversal	16.0	15.0
	Cashback / Unload	16.0	15.0
	Load	16.0	15.0
	Balance Inquiry	16.0	15.0
	Activate	16.0	15.0

Verifone Point NO

Operating Countries: Norway, Sweden, Denmark, Finland, Latvia, Lithuania, Estonia
Supported Terminals: Yomani/Xenta and CZAM/SMASH

Payment Terminal Function	Subtype	Xstore Release Version	EFTLink Release Version
Sign On		15.0	15.0
Sign Off		15.0	15.0
Card Payment	Payment by Credit	15.0	15.0
	Payment by Debit	15.0	15.0

Payment Terminal Function	Subtype	Xstore Release Version	EFTLink Release Version
Refund	Refund by Credit	15.0	15.0
	Refund by Debit	15.0	15.0
Reversal	Reversal of last transaction	15.0	15.0
Reconciliation	Reconciliation with Closure Functionality that is typically available via the hardware maintenance menu on the Xstore POS client, that is, uses the <ServiceRequest> / <ServiceResponse> API.	15.0	15.0
Card Swipe – non PCI cards	Get card data	15.0	15.0
Ticket Reprint (Direct)		-	15.0
EFTLink server compliant		-	15.0
IP PED		-	15.0

WorldLine (Banksys)

VIC Protocol

Operating Countries: Belgium, Luxembourg, Netherlands

Supported Terminals: Yomani, Xenta, CZAM/SMASH

Payment Terminal Function	Subtype	Xstore Release Version	EFTLink Release Version
Card Payment	Payment by Credit	15.0	15.0
	Payment by Debit	15.0	15.0
Refund	Refund by Credit	15.0	15.0
	Refund by Debit	15.0	15.0
Reconciliation	Reconciliation with Closure Functionality that is typically available via the hardware maintenance menu on the Xstore POS client, that is, uses the <ServiceRequest> / <ServiceResponse> API.	15.0	15.0

WorldPay

Operating Countries: UK

Supported Terminals: Verifone VX 820

Payment Terminal Function	Subtype	Xstore Release Version	EFTLink Release Version
Sign On		15.0	15.0
Card Payment	Payment by Credit	15.0	15.0
	Payment by Debit	15.0	15.0
Refund	Refund by Credit	15.0	15.0
	Refund by Debit	15.0	15.0
	Cardless Refund using Token	16.0	15.0
Reconciliation	Reconciliation with Closure Functionality that is typically available via the hardware maintenance menu on the Xstore POS client, that is, uses the <ServiceRequest> / <ServiceResponse> API.	15.0	15.0
DCC (currency conversion)	Foreign exchange tender amount, rate, currency code Functionality that is driven through the Core plugin by virtue of the <DeviceRequest> / <DeviceResponse> API.	15.0	15.0
Cashback	Cashback prompted on PED, return amount to POS	15.0	15.0
Token Support	Receive and record token for verified return	16.0	15.0
Combined EFT/Tax Receipts	Defer customer copy to <CardServiceResponse>	16.0	15.0
	Suppress Merchant copy for EMV Chip transactions	-	15.0
IP PED		-	15.0

A
Glossary

Card Circuit

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

Card IIN

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

DCC

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

JVM

Java Virtual Machine

PED

PIN entry device

PED Pooling

Where the EFTLink Server is used to manage a pool of PEDs to be shared between the POSs and allocated dynamically

Print Pooling

Where the EFTLink Server is used to manage a pool of printers to be shared between the POSs and allocated dynamically

Tender

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

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 Warehouse Management System (RWMS)
5. Oracle Retail Invoice Matching (ReIM)
6. Oracle Retail Price Management (RPM)
7. Oracle Retail Allocation
8. Oracle Retail Mobile Merchandising (ORMM)
9. Oracle Retail Customer Engagement (ORCE)
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 Predictive Application Server Batch Script Architecture (RPAS 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 Regular Price Optimization (RPO)
21. Oracle Retail Merchandise Financial Planning (MFP)
22. Oracle Retail Size Profile Optimization (SPO)
23. Oracle Retail Assortment Planning (AP)
24. Oracle Retail Item Planning (IP)
25. Oracle Retail Item Planning Configured for COE (IP COE)
26. Oracle Retail Advanced Inventory Planning (AIP)
27. Oracle Retail Integration Bus (RIB)
28. Oracle Retail Service Backbone (RSB)
29. Oracle Retail Financial Integration (ORFI)
30. Oracle Retail Bulk Data Integration (BDI)
31. Oracle Retail Integration Console (RIC)

32. Oracle Commerce Retail Extension Module (ORXM)
33. Oracle Retail Data Extractor for Merchandising
34. Oracle Retail Clearance Optimization Engine (COE)
35. Oracle Retail Analytic Parameter Calculator for Regular Price Optimization (APC-RPO)
36. Oracle Retail Insights, including Retail Merchandising Insights (previously Retail Merchandising Analytics) and Retail Customer Insights (previously Retail Customer Analytics)
37. Oracle Retail Order Broker