

**Oracle® Retail EFTLink**  
Core Configuration Guide  
Release 16.0  
E78218-07

March 2019

Oracle® Retail EFTLink Core Configuration Guide, Release 16.0

E78218-07

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Primary Authors: Matt Preston

Contributors: Tracy Gunston

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

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

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

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

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

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

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

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

---

## 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

|  |             |
|--|-------------|
| <b>Send Us Your Comments .....</b>                                 | <b>xi</b>   |
| <b>Preface .....</b>   | <b>xiii</b> |
| Audience .....   | xiii        |
| Documentation Accessibility .....                                  | xiii        |
| Related Documents .....  | xiii        |
| Customer Support .....   | xiii        |
| Review Patch Documentation .....                                   | xiv         |
| Improved Process for Oracle Retail Documentation Corrections ..... | xiv         |
| Oracle Retail Documentation on the Oracle Technology Network ..... | xiv         |
| Conventions .....  | xiv         |
| <b>1 Introduction .....</b>  | <b>1</b>    |
| <b>2 Adyen Core .....</b>  | <b>3</b>    |
| Disambiguation .....   | 3           |
| EFTLink General .....  | 3           |
| Minimum Version .....  | 3           |
| System Architecture .....  | 3           |
| Fileset .....  | 3           |
| Keystore .....   | 3           |
| Windows Operating Systems .....                                    | 3           |
| Password Encryption .....  | 4           |
| Windows Operating Systems .....                                    | 4           |
| Third Party .....  | 4           |
| Language .....   | 4           |
| Core Classname .....   | 5           |
| Configuration Settings .....                                       | 5           |
| Key Settings .....   | 5           |
| Secondary Settings .....   | 5           |
| Supported Functions .....  | 7           |
| <b>3 AJB FIPay .....</b>   | <b>9</b>    |
| Disambiguation .....   | 9           |
| EFTLink General .....  | 9           |
| Minimum Version .....  | 9           |
| System Architecture .....  | 9           |
| Fileset .....  | 9           |
| Third Party .....  | 10          |
| Language .....   | 10          |
| Core Classname .....   | 10          |
| Configuration Settings .....                                       | 10          |
| Key Settings .....   | 10          |

---

|  |           |
|--|-----------|
| Secondary Settings.....                | 11        |
| Supported Functions .....              | 12        |
| <b>4 Banksys Core .....</b>            | <b>13</b> |
| General Information .....              | 13        |
| EFTLink General .....                  | 13        |
| Minimum Version.....                   | 13        |
| System Architecture.....               | 13        |
| Fileset .....                          | 13        |
| Language.....                          | 13        |
| Core Classname.....                    | 13        |
| Configuration Settings .....           | 14        |
| Key Settings .....                     | 14        |
| Secondary Settings.....                | 14        |
| Supported Functions .....              | 14        |
| <b>5 Cayan Core .....</b>              | <b>15</b> |
| EFTLink General .....                  | 15        |
| Minimum Version.....                   | 15        |
| System Architecture.....               | 15        |
| Fileset .....                          | 15        |
| Account Information Entry.....         | 16        |
| Account Information Re-Encryption..... | 16        |
| Windows Operating Systems.....         | 16        |
| Core Classname.....                    | 16        |
| Configuration Settings .....           | 17        |
| Key Settings .....                     | 17        |
| Secondary Settings.....                | 17        |
| Administration Functions .....         | 17        |
| Supported Functions .....              | 19        |
| <b>6 CCVPOS (CCV ITS) .....</b>        | <b>20</b> |
| EFTLink General .....                  | 20        |
| Minimum Version.....                   | 20        |
| System Architecture.....               | 20        |
| Fileset .....                          | 20        |
| Language.....                          | 20        |
| Core Classname.....                    | 20        |
| Configuration Settings .....           | 21        |
| Key Settings .....                     | 21        |
| Secondary Settings.....                | 21        |
| Supported EFT Operations.....          | 22        |
| Administration Menu.....               | 22        |

---

|           |                                       |           |
|-----------|---------------------------------------|-----------|
| <b>7</b>  | <b>Ingenico .....</b>                 | <b>23</b> |
|           | EFTLink General .....                 | 23        |
|           | System Architecture.....              | 23        |
|           | Fileset .....                         | 23        |
|           | Language.....                         | 23        |
|           | EFTLink Configuration .....           | 23        |
|           | Configuration Settings .....          | 24        |
|           | Supported Functions .....             | 24        |
| <b>8</b>  | <b>Merchant Link.....</b>             | <b>25</b> |
|           | Disambiguation.....                   | 25        |
|           | EFTLink General .....                 | 25        |
|           | System Architecture.....              | 25        |
|           | Fileset .....                         | 25        |
|           | Core Classname.....                   | 25        |
|           | Configuration Settings .....          | 25        |
|           | Key Settings .....                    | 26        |
|           | Secondary Settings.....               | 26        |
|           | Supported Functions .....             | 26        |
|           | Administration Functions.....         | 26        |
| <b>9</b>  | <b>SixPay .....</b>                   | <b>29</b> |
|           | General Information .....             | 29        |
|           | EFTLink General .....                 | 29        |
|           | System Architecture.....              | 29        |
|           | Fileset .....                         | 29        |
|           | Language.....                         | 29        |
|           | Core Classname.....                   | 30        |
|           | Configuration Settings .....          | 30        |
|           | Key Configuration Settings .....      | 30        |
|           | Optional Configuration Settings.....  | 30        |
|           | Fixed Configuration Settings.....     | 31        |
|           | Other Information.....                | 31        |
|           | PED Identification/Selection .....    | 31        |
| <b>10</b> | <b>SolveConnect Core.....</b>         | <b>33</b> |
|           | General Information .....             | 33        |
|           | EFTLink General .....                 | 33        |
|           | System Architecture.....              | 33        |
|           | Fileset .....                         | 33        |
|           | Language.....                         | 33        |
|           | Core Classname.....                   | 33        |
|           | Configuration Settings .....          | 34        |
|           | Key Settings .....                    | 34        |
|           | Secondary Configuration Settings..... | 34        |

---

|   |           |
|---|-----------|
| Fixed Configuration Settings.....                 | 37        |
| Supported Functions .....                         | 38        |
| <b>11 TransaxEFT .....</b>                        | <b>39</b> |
| EFTLink General .....                             | 39        |
| Minimum Version.....                              | 39        |
| System Architecture.....                          | 39        |
| Fileset .....                                     | 39        |
| Translation .....                                 | 39        |
| Core Classname .....                              | 39        |
| Configuration Settings .....                      | 40        |
| Key Settings .....                                | 40        |
| Secondary Settings.....                           | 40        |
| Fixed Settings .....                              | 41        |
| Supported Functions .....                         | 41        |
| Maintenance Options .....                         | 42        |
| <b>12 VeriFone Ocius Sentinel .....</b>           | <b>43</b> |
| General Information .....                         | 43        |
| Overview.....                                     | 43        |
| System Architecture.....                          | 43        |
| Fileset.....                                      | 43        |
| Language.....                                     | 43        |
| Core Classname.....                               | 43        |
| Configuration Settings .....                      | 44        |
| Key Settings .....                                | 44        |
| Optional Configuration Settings.....              | 44        |
| Translating and Suppressing Status Messages ..... | 51        |
| Overriding Other Text Messages .....              | 51        |
| Positioning Dialogue Options .....                | 52        |
| XML Receipts.....                                 | 52        |
| Keystore.....                                     | 55        |
| Windows Operating Systems .....                   | 55        |
| Password Encryption .....                         | 55        |
| Windows Operating Systems.....                    | 55        |
| Supported Functions .....                         | 56        |
| <b>13 VeriFone Point Scandinavia.....</b>         | <b>57</b> |
| General Information .....                         | 57        |
| Disambiguation.....                               | 57        |
| EFTLink General .....                             | 57        |
| System Architecture.....                          | 57        |
| Fileset .....                                     | 57        |
| Language.....                                     | 57        |
| Core Classname.....                               | 58        |

---

|                                     |           |
|-------------------------------------|-----------|
| Configuration Settings .....        | 58        |
| Other Information .....             | 60        |
| Supported Functions .....           | 60        |
| <b>14 Verifone Point (US) .....</b> | <b>61</b> |
| EFTLink General .....               | 61        |
| Disambiguation .....                | 61        |
| Minimum Version .....               | 61        |
| System Architecture .....           | 61        |
| Fileset .....                       | 61        |
| Core Classname .....                | 61        |
| Configuration Settings .....        | 61        |
| Key Settings .....                  | 61        |
| Secondary Settings .....            | 62        |
| Administration Functions .....      | 62        |
| Supported Functions .....           | 63        |
| <b>15 WorldPay .....</b>            | <b>65</b> |
| General Information .....           | 65        |
| Overview .....                      | 65        |
| System Architecture .....           | 65        |
| Fileset .....                       | 65        |
| Language .....                      | 65        |
| Core Classname .....                | 65        |
| Configuration Settings .....        | 65        |
| Other Information .....             | 68        |
| Additional Functions .....          | 68        |
| Integration Notes .....             | 68        |



---

---

# Send Us Your Comments

Oracle Retail EFTLink, Core 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](mailto: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](http://www.oracle.com).

---

---

# Preface

This *Oracle Retail EFTLink Core Configuration Guide* describes the requirements and procedures to set up EFTLink to interface between the specific POS and the selected EFT payment system

## Audience

This *Oracle Retail EFTLink Core Configuration 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 Framework Installation and Configuration Guide*
- *Oracle Retail EFTLink Security 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
```

---

---

# Introduction

After installing EFTLink from the *Oracle Retail EFTLink Framework Installation and Configuration Guide*, and as part of that selected a core (step 2 in that document), the implementer will need to configure the specific core with the required settings to allow the POS to communicate with the selected EFT System. This guide consists of separate sections for each available core; go to the pertinent section for each core to be installed.

---

---

**Note:** Also refer to the *Oracle Retail EFTLink Security Guide* for core specific actions to ensure secure configuration.

---

---



---

---

# Adyen Core

## Disambiguation

This core implementation is for use with Adyen JNI wrapper with communication based on a socket or serial protocol, implemented internally within the JNI, to the terminal.

## EFTLink General

See also the *Oracle Retail EFTLink Framework Installation and Configuration Guide*.

This document assumes static EFTLink configuration. When deploying with a POS that supports dynamic configuration, all property settings referred to below should be set on the POS, and not directly into local property files.

## Minimum Version

The Adyen interface requires a minimum EFTLink version of v15.0

## System Architecture

EFTLink connects to Adyen's PED, via JNI wrapper.

---

---

**Note:** This document does not cover installation of Adyen software

---

---

## Fileset

In addition to standard EFTLink files, Adyen uses:

- `cores/Adyen/AdyenCore.jar` – executable code for the Adyen EFTLink core.
- `adyen.properties` – configuration settings to specify which features are enabled and to define communication parameters for the interface with the EFT payment system.
- `data/adyen.keystore` – keystore file to encrypt a password in `adyen.properties`, this file need to be generated at installation. Please see next section for details.

## Keystore

The encryption key must be generated and stored in a keystore. To achieve this, the following steps must be followed:

## Windows Operating Systems

- Open a command prompt, and change directory to the `eftlink` location.
  - Type `encrypt.bat -k <keystore name> <properties file>`  
For example, `encrypt.bat -k adyen.keystore adyen.properties`.
- Keystore file will be generated and stored in the `data` directory

## Password Encryption

The password within the `adyen.properties` file needs to be encrypted. To achieve this, the following steps must be followed:

### Windows Operating Systems

To encrypt a password; open a command prompt and change directory to eftlink's location.

- Type `encrypt.bat -e <keystore name> <properties> <password>`.  
For example `encrypt.bat -e adyen.keystore adyen.properties` [followed by the required password as a final parameter].
- Password and initialization vector will be outputted to the console.  
Copy and paste it to `adyen.password` and `adyen.password.iv` in `adyen.properties`.

To re-encrypt a password with new encryption settings; open a command prompt and change directory to eftlink's location.

- Type `encrypt.bat -r <keystore name> <properties> <encrypted password> <previous initialization vector> <keygen type> <cipher type> <key size> <iterations>`.  
For example, `encrypt.bat -r adyen.keystore adyen.properties [Encrypted password] [Encrypted password iv] AES AES/CBC/PKCS5Padding 128 10000`.
- Re-encryption uses existing crypto settings in the properties file to decrypt the password. Once the password is decrypted, a new keystore file is generated using the new crypto parameters specified at the command line and the new encrypted password / initialization vector is generated. /
- When using AES algorithm with a keysize that is greater than 128, you may get `java.security.InvalidKeyException: Illegal key size or default parameters`. If so, Additional Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files will need to be downloaded and extracted to `%JAVA_HOME%/jre/lib/security/`

## Third Party

---

---

Note: Critically important

---

---

The following file is also needed, not supplied by Oracle:

`POS_JNI32.jar/POS_JNI64.jar` is a JNI wrapper supplied by Adyen to allow communication to Adyen's PED.

This should be placed in `cores\Adyen` alongside `AdyenCore.jar`. Use appropriate version according to VM environment, `POS_JNI32.jar` for 32-bit and `POS_JNI64.jar` for 64-bit.

## Language

There are translation files in `AydenCore.jar`. These should not need to be modified, but if a translation needs to be changed, they can be extracted to the base eftlink folder.

`LangEN_Ayden.properties`

The file in use follows the language setting for EFTLink itself, defined in `EftlinkConfig.properties` for example, `DisplayLanguage = EN`

Additional files could be added for other supported languages for the small set of translations required by the Adyen core.

---

## Core Classname

The following should have been set in the `EftlinkConfig.properties` file by `installcore.bat` or `installcore.sh`

```
EPSCore0 = manito.eft.adyen.AdyenCore
```

## Configuration Settings

The full set of configuration properties is defined and commented in `adyen.properties`.

## Key Settings

Settings that may be different for all POSs.

| Setting                             | Description  | Example  |
|-------------------------------------|--|--|
| <code>adyen.environment</code>      | Live or Test environment. Default is Test.                       | <code>adyen.environment = Live</code>            |
| <code>adyen.merchant.account</code> | Merchant account code provided by Adyen.                         | <code>adyen.merchant.account = OracleTest</code> |
| <code>adyen.username</code>         | Username provided by Adyen.                                      | <code>adyen.username = ws_371398</code>          |
| <code>adyen.password</code>         | Encrypted password, see password encryption section for details. | <code>host = 10.0.0.99</code>                    |
| <code>ped.address</code>            | IP address of the PED. If serial ped then com port number.       | <code>ped.address = 10.0.0.99</code>             |

## Secondary Settings

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

| Setting                         | Description   | Default | Example   |
|---------------------------------|---|---------|---|
| <code>ped.name</code>           | Any symbolic name of the PED.   |         | <code>ped.name = VX680_01</code>  |
| <code>merchant.reference</code> | Unique merchant reference.  |         | <code>merchant.reference = Merchant_A</code>  |
| <code>tender.options</code>     | Specify tender options to be used. Currently supported options are:<br><code>AskGratuity, AttendantActionHandler, BypassPin, DontPrintReceipt, EnableMagstripeFallback, Error, ForcedDecline, ForcedOnline, GetAdditionalData, KeyedCardDetailsHandler, KeyedEntry, NoCTLS, NoProcess, ReceiptHandler, UNKNOWN</code> |         | <code>tender.options = GetAdditionalData, ReceiptHandler, AttendantActionHandler</code> |



| Setting                          | Description  | Default | Example                                    |
|----------------------------------|--|---------|--|
| tokenized.refund                 | Enables refund by token if set to true or auto. If set to false, standard refund will always be performed. If set to auto, tokenized refund will be performed if a token is supplied in the request otherwise standard refund will be used.  | auto    | tokenized.refund = auto                    |
| combine.receipt                  | When <code>combine.receipt</code> is true, sets which line number to suppress  |         | combine.receipt = true                     |
| combine.receipt.suppress.lines   | When <code>combine.receipt</code> is true, sets which line number to suppress  |         | combine.receipt.suppress.lines = 1,2,3,4,5 |
| combine.receipt.suppress.strings | When <code>combine.receipt</code> is true, sets which line to suppress when strings are matched.   |         | combine.receipt.suppress.lines = Date,Time |
| pos.id.override                  | Overrides POS ID from the POS with a specify ID. Should be used in PED POOL mode.  |         | pos.id.override = 10                       |
| cashback                         | Enables cashback Following options are supported:<br>off - no cashback<br>pos - POS will prompt for cashback options via DeviceRequest<br>ped - PED will prompt for cashback options via its display   |         | cashback = ped                             |
| print.all.receiptSets            | When set to true, enables all receipts sent from Adyen to be printed. When set to false, prints only the latest receipt set.   |         | print.all.receiptSets = false              |
| crypto.keygenType                | Sets keygen algorithm type.  |         | crypto.keygenType = AES                    |
| crypto.cipherType                | Sets cipher algorithm type.  |         | crypto.cipherType = AES                    |
| crypto.keySize                   | Sets size of the keystore.<br>Note that when keysize is greater than 128, you may get <code>java.security.InvalidKeyException: Illegal key size or default parameters</code> . If this happens you will need to download additional Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files and extract those files to <code>\${java.home}/jre/lib/security/</code> |         | crypto.keySize = 128                       |
| crypto.iterations                | Sets number of iterations.   |         | crypto.iterations = 1000                   |

---

## Supported Functions

Below is a list of supported functionalities of the interface to Adyen. Some functions provided by Adyen, such as Loyalty, Giftcard, Cashback etc. are not implemented in this release because of the business requirement.

| Function         | Description   |
|------------------|---|
| Payment          | EFTLink sends payment requests to Adyen. Adyen will return a response message with unformatted receipt strings for customer and/or merchant receipts.<br>If successful, appropriate receipts will be printed at the end of transaction. |
| Reversal         | EFTLink sends reversal requests to Adyen. This will reverse a transaction specified by the transaction number, found on the receipt, which must be captured by the POS and pass on to EFTLink.  |
| Refund           | EFTLink sends refund requests to Adyen. This will refund a transaction with specified amount.   |
| Tokenized Refund | EFTLink sends refund requests to Adyen. This will refund a transaction with specified token id.   |



---

---

## AJB FIPay

This FIPay implementation is for use with AJB FIPay software with communication via TCP/IP based on a proprietary socket protocol. It should be read in conjunction with the *Oracle Retail EFTLink Framework Installation and Configuration Guide*.

### Disambiguation

This FIPay implementation is for use with any compatible terminal that has AJB firmware installed, with communication based on a socket protocol.

### EFTLink General

This document assumes static EFTLink configuration. When deploying with a POS that supports dynamic configuration, all property settings referred to below should be set on the POS, and not directly into local property files.

### Minimum Version

The FIPay interface requires a minimum EFTLink version of 1.1.118

### System Architecture

EFTLink connects directly to the terminal using a proprietary socket protocol.

---

---

**Note:** This document does not cover installation of AJB software

---

---

### Fileset

In addition to standard EFTLink files, FIPay uses:

- `FIPayCore.jar` - executable code for the FIPay EFTLink core.
- `fipay.properties` - configuration settings to specify which features are enabled and to define communication parameters for the interface with the EFT payment system.
- `Giftcard.properties` - Giftcard specific properties.
- `LangEN_FIPay.properties` - English language translation file, alternative translation may be used by adding a new file and change the 2-character language code and translation text as appropriate.
- `AJBComm.jar` - API supplied by AJB to allow communication to the terminal.

---

---

**Note:** If the POS supports dynamic configuration, properties can be set there instead of in `fipay.properties`.

---

---

## Third Party

Note: Critically important

The following file is also needed, not supplied by Oracle: `AJBComm.jar`. This is an API supplied by AJB to allow communication to FIPay software. It should be placed in `cores\FIPay` alongside `FIPayCore.jar`.

## Language

There are translation files in `FIPayCore.jar`. These should not need to be modified, but if a translation needs to be changed, they can be extracted to the base `eftlink` folder.

`LangEN_FIPay.properties`

`LangFR_FIPay.properties`

The language used will follow the language set in the EFTLink framework; see the *Oracle Retail EFTLink Framework Installation and Configuration Guide*, EFTLink General Information, Translation section.

`EftlinkConfig.properties`

`DisplayLanguage = EN`

Possible values are: EN, FR

Additional files could be added for other EFTLink supported languages for the small set of translations required by the FIPay core.

## Core Classname

`manito.eft.ajb.FIPayCore`

This should be set as `EPSCore<x>` in `eftlinkconfig.properties`.

## Configuration Settings

The full set of configuration properties is defined and commented in `fipay.properties`.

## Key Settings

Settings that may be different for each POS/PED.

| Setting                   | Description                               | Example                             |
|---------------------------|---|-------------------------------------|
| <code>ip.address</code>   | Terminal address IP address               | <code>ip.address = localhost</code> |
| <code>store.number</code> | The unique store number allocated by AJB. | <code>store.number = 100</code>     |

## Secondary Settings

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

| Setting                   | Description  | Default  | Example                              |
|---------------------------|--|--|--------------------------------------|
| ip.port                   | IP port number   | 24900  | ip.port = 24900                      |
| creditdebit.prompt        | Credit/Debit prompt, controls whether to prompt operator for the card type (debit or credit), a specific terminal may have this built-in so this property maybe turned off (set to false).   | False meaning FiPay will determine card type automatically | creditdebit.prompt = false           |
| response.timeout          | FiPay response timeout, specify the number of seconds to wait for response from FiPay.   | 120  | response.timeout = 120               |
| pos.validate.swipe        | Card validation prompt, controls whether to continue with the payment for this card. The prompt will display the card type.  | false  | pos.validate.swipe = false           |
| electronic.signature      | Enable electronic signature capture, if false signature prompt will appear after receipts are printed.   | true   | electronic.signature = true          |
| enable.signature.logging  | Enable logging of signature data (for debugging purposes ONLY).<br><b>Note:</b> This should be enabled for debugging purposes only. As soon as the debugging is complete, set back to false. | false  | electronic.signature.logging = false |
| enable.emv.initialization | Enable emv transaction processing, when enabled, it will send an 'initDebit' command to FiPay at pos logon, an admin option is also available to allow adhoc initialization.                 | false  | enable.emv.initialization = false    |
| emvkeys.provided          | EMV initialisation keys 'EMV_KEYS.DAT' which resides in FIPAYEPS, set to true if this file exists. Ignored if 'enable.emv.initialization' is false.  | false  | emvkeys.provided=false               |
| enable.tokenization       | Enable tokenization for refund.  | false  | enable.tokenization=false            |
| currency.symbol           | Currency symbol for customer display. If set to 'default', symbol base on operating system regional setting will be used.  | \$   | currency.symbol=\$                   |
| combine.receipt           | Turn on/off POS combine receipt, default true  | true   | combine.receipt=true                 |

| Setting                          | Description   | Default | Example   |
|----------------------------------|---|---------|---|
| combine.receipt.suppress.lines   | When combine receipt is true, set which line to suppress.   |         | combine.receipt.suppress.lines=1,2,3,4                                  |
| combine.receipt.suppress.strings | When combine receipt is true, set what line to suppress when strings are matched  |         | combine.receipt.suppress.strings=DATE,DCC Not Available                 |
| giftcard.handler                 | Giftcard handler, fully qualified class name  |         | giftcard.handler = manito.eft.ajb.giftcard.StandardFiPayGiftCardHandler |
| giftcard.provider                | Giftcard provider, fully qualified class name. Possible values are: manito.eft.ajb.giftcard.FiPayBlackhawk, manito.eft.ajb.giftcard.FiPaySVS, manito.eft.ajb.giftcard.FiPayGiveX, manito.eft.ajb.giftcard.FiPayInComm, manito.eft.ajb.giftcard.FiPayValueLink |         | giftcard.provider =   |

## Supported Functions

The following operations are supported by this implementation of the AJB FiPay interface.

| Function                    | Description   |
|-----------------------------|---|
| Payment                     | EFTLink sends payment requests to AJB FiPay. AJB will return a response message with formatted receipt strings for customer and/or merchant receipts. In an event of referral where authorization cannot be obtained online then a prompt for authorization code will appear; the authorization code must be obtained via telephone and entered here.<br><br>If successful, appropriate receipts will be printed at the end of transaction. |
| Reversal                    | EFTLink sends reversal requests to AJB FiPay. This will reverse a transaction specified by the transaction number, found on the receipt, which must be captured by the POS and passed on to EFTLink.  |
| Refund                      | EFTLink sends refund requests to AJB FiPay. This will refund a transaction with specified amount.   |
| Reconciliation / Settlement | This is not supported directly by AJB FiPay via TCP/IP request; instead a batch script supplied by AJB must be used. This can be set up to run automatically at a specific time or on-demand at user's discretion.  |

---

---

# Banksys Core

## General Information

This document covers EFTLink Integration with **Banksys VIC** Payment Systems. It should be read in conjunction with the *Oracle Retail EFTLink Framework Installation and Configuration Guide*.

## EFTLink General

This document assumes static EFTLink configuration. When deploying with a POS that supports dynamic configuration, all property settings referred to below should be set on the POS, and not directly into local property files.

## Minimum Version

The **Banksys** interface requires:

- Aa minimum EFTLink version of 15.0.
- Java 1.6 or later.

## System Architecture

EFTLink connects directly to the payment terminal using serial communication using the VIC protocol.

## Fileset

In addition to standard EFTLink files:

- `cores/Banksys/banksysvic.jar` - executable code for the core
- `lib/RXTXcomm.jar` - Serial comms library.
- `lib/rxtxParallel.dll` - Windows DLL library for RXTXcomm.jar.
- `lib/rxtxSerial.dll` - Windows DLL library for RXTXcomm.jar.

## Language

There are translation files in `banksysvic.jar`

```
EftlinkConfig.properties
```

```
DisplayLanguage = EN
```

Instead the translations selected will follow the value of a Banksys specific setting `LANGUAGE`, see secondary setting below. Possible languages are English, French and Dutch.

## Core Classname

```
com.torexretail.eftlink.core.vic.VicCore
```

This should be set as `EPSCore0 = com.torexretail.eftlink.core.vic.VicCore` in `EftlinkConfig.properties` by `installcore.bat` or `installcore.sh`



## Configuration Settings

The banksys core does not have a dedicated core property file, instead the properties are defined by entries in the framework property file, `EftlinkConfig.properties`. These entries are not present by default, so a `readme.txt` in the `cores/Banksys` folder contains commented examples of these entries, which can be pasted into `EftlinkConfig.properties`

## Key Settings

Settings that may be different for each POS/PED.

| Setting      | Description   | Default | Example             |
|--------------|---|---------|---------------------|
| VIC_PORTNAME | Serial port name.   |         | VIC_PORTNAME = COM1 |
| LANGUAGE     | Language code, options: en = English, fr = French, nl = Dutch | en      | LANGUAGE = en       |
| COUNTRY      | Country code.   | UK      | COUNTRY = UK        |

## Secondary Settings

Settings that are normally correct at their default values, but can be overridden if necessary.

| Setting              | Description   | Default         | Example                      |
|----------------------|---|-----------------|------------------------------|
| VIC_AUTOLOGON        | Automatically perform a log-on at initialization.   | false           | VIC_AUTOLOGON = false        |
| VIC_MANDATORYVOUCHER | If receipt data is not available, receipt voucher will be created using the data from the response. | false           | VIC_MANDATORYVOUCHER = false |
| VIC_PRINTWIDTH       | Receipt print width.  | 38 (characters) | VIC_PRINTWIDTH = 38          |

## Supported Functions

The following operations are supported by this implementation of the Banksys interface:

### Payment

Sends payment request to the Banksys terminal. The terminal will return a response message with formatted receipt strings for customer and/or merchant receipts.

If successful, appropriate receipts will be printed at the end of transaction.

---

---

# Cayan Core

This Cayan implementation is for use with Genius terminals in the US, with communication based on a webservice protocol.

## EFTLink General

See also the EFTLink general deployment guide if not already familiar with EFTLink.

This document assumes static EFTLink configuration. When deploying with a POS that supports dynamic configuration, all property settings referred to below should be set on the POS, and not directly into local property files.

## Minimum Version

The Cayan interface requires a minimum EFTLink version 16

## System Architecture

Cayan Genius is deployed as an intelligent terminal. EFTLink connects directly to the terminal using a proprietary web services protocol.

Genius 5.0 and later versions supports a HTTPS interface in addition to its traditional HTTP interface. Only the protocol scheme (https vs. http) and port (8443 vs 8000) differ. The cayancore can communicate with the Genius device using TLS to secure the connection. The terminal will generate appropriate certificates as required in order to serve the TLS connection, and all certificates generated by the terminal will be signed by the Cayan CA.

The cayan certificate is automatically stored upon startup in the file `cayan.public.jks`

To enable TLS in `cayan.properties`, change all the `http.action` entries containing <http://cedIp:cedPort> into <https://cedIp:cedPort> and set `ced.port=8443`

## Fileset

In addition to standard EFTLink files:

- `cayancore.jar` – executable code for the Cayan EFTLink core.
- `cayanTA.crt` – cayan root certificate
- `cayan.properties` – configuration settings to specify which features are enabled and to define communication parameters for the interface with the terminal.
- `langEN_cayan.properties` – English translation file for the Cayan core
- `cayanruntime.properties` – core logging settings that are automatically reloaded at runtime (checked every 10 seconds)
- `cayandynamic.properties` – merchant specific details that can be accessed through the administration functions
- `cayan_receipt.properties` – links a receipt template file to a ReceiptType XML element
- `cayan_giftadd_receipt`, `cayan_giftbalance_receipt`, `cayan_payment_receipt`, `cayan_refund_receipt`, `cayan_reversal_receipt` – customer configurable receipt template files

### Runtime files

- `cayan.public.jks` – keystore file containing the cayan root certificate to allow TLS communication
- `cayan.secure` – storage file for the random encryption key that is used to protect merchant information

## Account Information Entry

At initial software startup, a keystore is created for encryption information and the Cayan certificate is placed into a second keystore. Account information is added to the EFTLink system via the EFTLink admin menus. Five parameters are required to be entered via the admin function:

- Account Name
- Account Software Key
- Site Identifier
- Account DBA
- Terminal Identifier

Both the Account Name and Account Software Key are automatically encrypted. All 5 parameters are held in the `cayandynamic.properties` file.

See the [Administration Functions](#) section below for entry of the parameters.

## Account Information Re-Encryption

The password within the `cayandynamic.properties` file needs to be encrypted. To achieve this, the following steps must be followed:

### Windows Operating Systems

To re-encrypt a password with new encryption settings; open a command prompt and change directory to eftlink's location.

- Type `encrypt.bat -g <keystore name> <properties> <certificate> <dyanamicProperties> {<Colon-Separated List of Properties>} <keygenType> <cipherType> <keySize> <iterations>`

For example, `encrypt.bat -g cayan.secure cayan.properties cayan.public.jks cayandynamic.properties {merchant.name:merchant.key} AES AES 128 10000`

- Re-encryption uses existing crypto settings in the properties file to decrypt the password. Once the password is decrypted, a new keystore file is generated using the new crypto parameters specified at the command line and the new encrypted password / initialization vector is generated.
- When using AES algorithm with keysize is greater than 128, you may get **java.security.InvalidKeyException: Illegal key size or default parameters**. If so Additional Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files will need to be downloaded and extracted to `${java.home}/jre/lib/security/`

## Core Classname

`manito.eft.cayan.CayanCore`

This should be set as `EPSCore<x>` in `eftlinkconfig.properties`.

---

## Configuration Settings

The full set of configuration properties is defined and commented in `cayan.properties`.

### Key Settings

Settings that may be different for each POS/PED.

| Setting                | Description  | Example                                   |
|------------------------|--|---|
| Terminal address       | IP of Genius terminal.   | <code>ced.ip =</code>                     |
| Receipt handling       | Separate EFT receipts or EFT receipt as part of the regular POS receipt. | <code>EmbeddedReceipt=false</code>        |
| Signature Verification | Enable/Disable signature verification dialog                             | <code>SignatureVerification =false</code> |
| Reversal Failure       | Enable/Disable reversal failure dialog                                   | <code>ReversalDialog =false</code>        |

### Secondary Settings

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

| Setting                   | Description                         | Default                           | Example                         |
|---------------------------|-------------------------------------|-----------------------------------|---------------------------------|
| Terminal address          | Port number                         | 8080 for http and 8443 for https. | <code>ced.port =</code>         |
| Timeout                   | Overall response timeout in seconds | 600                               | <code>ced.get.timeout =</code>  |
| Signature display scaling | Signature display scaling           | 3                                 | <code>SignatureScaling =</code> |

## Administration Functions

The terminal has some administration/maintenance functions. These are normally invoked from a dedicated "EFT Maintenance" button, but if this is not available, they could be accessed by an engineer using the EFTLink built-in test harness.

EFTLink uses DeviceProxy messages to display input prompts on the POS to manage these functions.

Cayan will provide the merchant credentials that are required to setup the connection with the Cayan host. The information consists of five elements: Name, Key, SiteID, DBA and TerminalID

These credentials must be entered through the administration functions. The information is stored in the file `cayandynamic.properties`. The fields Name and Key are stored in an encrypted form. For each POS system, the cayancore will create a random encryption key to protect sensitive information. The encryption key itself is stored in the file `cayan.secure` using an EFTLink specific encryption algorithm.

Cayan has created an Oracle account for testing purposes. To connect to the Cayan host from non-US IP addresses, a 'WhitelistRequest' document containing the static IP of the Genius terminal must be sent to Cayan first. It typically takes 2-3 business days for Cayan security to review and then IT to process.

| <b>Operation</b>     | <b>Description</b>  |
|----------------------|---|
| Merchant Name        | This operation allows the technician/cashier to enter the merchant name and store it encrypted in <code>cayandynamic.properties</code> .      |
| Merchant Key         | This operation allows the technician/cashier to enter the merchant key and store it encrypted in <code>cayandynamic.properties</code> .       |
| Merchant Site ID     | This operation allows the technician/cashier to enter the merchant site identifier and store it in <code>cayandynamic.properties</code> .     |
| Merchant DBA         | This operation allows the technician/cashier to enter the merchant dba and store it in <code>cayandynamic.properties</code> .                 |
| Merchant Terminal ID | This operation allows the technician/cashier to enter the merchant terminal identifier and store it in <code>cayandynamic.properties</code> . |

---

## Supported Functions

Below is a list of supported functionalities of the interface to Cayan.

| Function                 | Description  |
|--------------------------|--|
| Payment                  | Sends payment request to the terminal. Terminal will return a response message with receipt strings.   |
| Reversal                 | Sends reversal request to the terminal. This will reverse a transaction specified by the transaction number, found on the receipt, which must be captured by the POS and pass on to EFTLink.   |
| Refund                   | Sends refund request to the terminal. This will refund a transaction with specified amount.  |
| Sale State Notifications | Sends line items through to the device so the customer display can be updated in line with the POS.  |
| SVC Payment              | Sends a Gift or Merchandise credit card payment request to the terminal. If there are not enough funds available, only the funds available will be deducted. The POS client will have to settle the transaction with another tender in this scenario.                          |
| SVC Activate             | Sends a Gift or Merchandise credit card activation request to the terminal.  |
| SVC Deactivate           | Sends a Gift or Merchandise credit card deactivation request to the terminal. The account is disabled after this as the request is intended to be used for lost or stolen cards. It is not possible to use the card or account once this request has been issued and accepted. |
| SVC Add Value            | Sends a Gift or Merchandise credit card add value request to the terminal. This will only add value to an account that has been activated.   |
| SVC Balance Enquiry      | Sends a Gift or Merchandise credit card balance enquiry request to the terminal.   |
| SVC Unload (Cashout)     | Sends a Gift or Merchandise credit card cash out request to the terminal. All funds are deducted from the account and the cash back amount is returned to the POS. The account is not deactivated as part of this process.   |

---

---

# CCVPOS (CCV ITS)

This CCVPOS implementation is for use with payment terminals that support the CCV ITS interface, with communication based on a socket/XML protocol.

## EFTLink General

This document assumes static EFTLink configuration. When deploying with a POS that supports dynamic configuration, all property settings referred to below should be set on the POS, and not directly into local property files. It should be read in conjunction with the *Oracle Retail EFTLink Framework Installation and Configuration Guide*.

### Minimum Version

The CCVPOS interface requires a minimum EFTLink version of v15.0.

## System Architecture

CCV ITS is deployed as an intelligent terminal. EFTLink connects directly to the terminal via TCP/IP using a socket/XML protocol derived from the OPI/IFSF standard.

## Fileset

In addition to standard EFTLink files, CCVPOS uses:

- `cores/ccvpos/ccvposcore.jar` - executable code for the CCVPOS EFTLink core.
- `CcvposConfig.properties` - configuration settings to specify which features are enabled and to define communication parameters for the interface with the EFT terminal.

## Language

There are translation files in `ccvposcore.jar`, that should not need to be modified, but if a translation needs to be changed, they can be extracted to the base `eftlink` folder.

`LangEN_CCV.properties`

`LangFR_CCV.properties`

`LangNL_CCV.properties`

The language used will follow the language set in the EFTLink framework; see the *Oracle Retail EFTLink Framework Installation and Configuration Guide*, EFTLink General Information, Translation section.

`EftlinkConfig.properties`

`DisplayLanguage = EN`

Possible values are: EN, FR, NL

## Core Classname

The following should have been set in the `EftlinkConfig.properties` file by `installcore.bat` or `installcore.sh`

`EPSCore0 = manito.eft.ccvpos.CcvPosCore`

---

## Configuration Settings

Settings are defined in `pointus.properties`.

### Key Settings

Settings that may be different for each POS/PED.

| Setting          | Description  | Default | Example                  |
|------------------|--|---------|--------------------------|
| CCVServerIP      | IP address of the CCV terminal.  |         | CCVServerIP = 10.0.0.99  |
| CCVWorkstationId | Name by which the local POS is identified by the terminal and/or host. | POS01   | CCVWorkstationId = POS01 |

### Secondary Settings

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

| Setting            | Description   | Default | Example                   |
|--------------------|---|---------|---------------------------|
| CCVChannel1        | TCP/IP port used by EFTLink to allow the CCV terminal to connect back to the POS.   | 4102    | CCVChannel1 = 4102        |
| CCVResponseTimeout | Time allowed in seconds for the transaction to complete at the terminal. This must be set longer than maximum terminal activity timeout, which is generally 5 minutes. This needs to be long enough to cover all customer interaction and host authorization. | 330     | CCVResponseTimeout = 330  |
| EmbedCustomerText  | Option to embed the customer EFT voucher within the POS receipt to save paper.<br><b>Note:</b> This goes against the CCV general rule of guaranteeing customer printout, so this feature should not be enabled without prior approval by CCV.                 | false   | EmbedCustomerText = false |
| EmbedJournalText   | Option to return the merchant EFT voucher to the POS as part of the payment response, rather than as a direct print request, so that it can be store in an Electronic Journal.  | false   | EmbedJournalText = false  |



| Setting         | Description  | Default | Example   |
|-----------------|--|---------|---|
| ManagementMenuX | Specifies which administration function to show at position 'n' of the menu and sets a label for it. See section on Administration Menu. |         | ManagementMenu1 = Reprint Last Transaction, REPRINT_LAST_TICKET<br>ManagementMenu2 = Reconciliation - shift totals, SHIFT_TOTALS<br>ManagementMenu3 = Reconciliation - close shift, CLOSE_SHIFT<br>ManagementMenu4 = List card types, GET_CARD_CIRCUITS<br>ManagementMenu5 = Version, VERSION<br>ManagementMenu6 = Cancel, CANCEL |

## Supported EFT Operations

The following operations are supported by this implementation of the CCV ITS interface.

### Payment / Refund

Payment and refund by credit/debit card.

### Reversal / Void

Payment/refund transactions can be cancelled by request from the POS.

### Reconciliation / Settlement

If required, EFT batch management and reporting can be managed via the Administration Menu. (See below).

## Administration Menu

The CCV ITS interface requires some administration/maintenance operations. These are normally invoked from a dedicated "EFT Maintenance" button at the POS, though the content of the display screen then presented to the operator is controlled by EFTLink.

### Reprint Last Transaction

Print a copy-receipt of the most recent transaction. This can be used to recover payment details in the event of a system failure.

### List Card Types

Print a report of the card type supported by the terminal.

### Reconciliation - Shift Totals

Print a non-closing shift/batch report.

### Reconciliation - Close Shift

Close the current shift/batch.

---

---

# Ingenico

This document section covers EFTLink Integration with Ingenico Payment Systems. It should be read in conjunction with the *Oracle Retail EFTLink Framework Installation and Configuration Guide*.

## EFTLink General

This document assumes static EFTLink configuration. When deploying with a POS that supports dynamic configuration, all property settings referred to below should be set on the POS, and not directly into local property files.

### Minimum Version

The Ingenico interface requires a minimum EFTLink version of 15.0.

## System Architecture

EFTLink connects to the payment system using a proprietary socket protocol. The Ingenico EPS runs as a software package called C3 installed on the POS PC.

---

---

**Note:** This document does not cover the installation of C3.

---

---

## Fileset

The following files are used:

- `cores/Ingenico/ingenicoCore.jar`
- `ingenico.properties`

## Language

There are translation files in `injenicoCore.jar`, that should not need to be modified, but if a translation needs to be changed, they can be extracted to the base `eftlink` folder.

```
LangEN_Ingenico.properties
LangES_Ingenico.properties
LangFR_Ingenico.properties
```

The language used will follow the language set in the EFTLink framework; see the *Oracle Retail EFTLink Framework Installation and Configuration Guide*, EFTLink General Information, Translation section

```
EftlinkConfig.properties
```

```
DisplayLanguage = EN
```

Possible values are: EN, ES, FR

## EFTLink Configuration

The following line should have been set in the `EftlinkConfig.properties` file by running `installcore.bat` or `installcore.sh`:

```
EPSCore0 = manito.eft.ingenico.IngenicoCore
```

## Configuration Settings

The core is configured via properties contained in the `ingenico.properties` file, which should have been copied from the `cores\Ingenico` folder to the base `eftlink` folder by `installcore.bat` or `installcore.sh`.

The available settings are listed below.

| Setting                        | Description   | Default | Example  |
|--------------------------------|---|---------|--|
| <code>c3path</code>            | Path to where C3 is located, this enables C3 to be initiated automatically by EFTLink. If left blank then the user must ensure that C3 is already running before launching EFTLink.                                       |         | <code>c3path = C:/Program Files/Ingenico/C3Generic/bin/c3inet.exe</code>                               |
| <code>ip.address</code>        | The IP address of C3, default 127.0.0.1.  |         | <code>ip.address = 10.0.0.5</code>   |
| <code>ip.port</code>           | The IP port of C3.  | 9518    | <code>ip.port = 9518</code>  |
| <code>confirmation_list</code> | A list of strings separated by a comma [,] which determines whether a user acknowledgement is required.<br><b>Note:</b> This list may require modification before it can be used in a Spanish configured Ingenico system. |         | <code>confirmation_list = APPEL PHONIE, &lt;&gt;, VALIDEZ, Montant, PAN, FIN VAL, APPEL TELECOL</code> |
| <code>comms_timeout</code>     | The timeout in seconds between EFTLink and C3.  |         | <code>comms_timeout = 120</code>   |
| <code>totalisation_type</code> | The report type when doing reconciliation - not currently used.   |         | <code>totalisation_type = C</code>   |
| <code>default_currency</code>  | The currency being used, defined by using the ISO 4217 numerical code, for example Euro 978, US Dollar 840, Sterling 826.   |         | <code>default_currency = 978</code>  |

## Supported Functions

The following operations are supported by this implementation of the Ingenico C3 interface.

- Logon and logoff (at the beginning and end of a shift or trading period)
- Sale (2-stage)
- Refund
- Maintenance
- Gift card/private card
- Cheque
- Customer receipt re-print (via the maintenance menu)

---

---

# Merchant Link

## Disambiguation

This Merchant Link implementation is for use with any compatible terminal that has Merchant Link firmware installed, with communication based on a socket protocol.

## EFTLink General

This document assumes static EFTLink configuration. When deploying with a POS that supports dynamic configuration, all property settings referred to below should be set on the POS, and not directly into local property files.

### Minimum Version

The Merchant Link interface requires a minimum EFTLink version of 1.1.124.

## System Architecture

Merchant Link is deployed as an intelligent terminal. EFTLink connects directly to the PoslynxMINI device which in turn connects to the Verifone MX925 device using a proprietary socket/XML protocol. The PoslynxMINI device acts as a message broker to the MX925 device. The MX925 device will need to know the PoslynxMINI device address. When setting the system up for the first time it is best to contact the Merchant Link technician who will be able to talk through the process.

## Fileset

In addition to standard EFTLink files:

- `poslynxcore.jar` - executable code for the Merchant Link EFTLink core.
- `poslynx.properties` - configuration settings to specify which features are enabled and to define communication parameters for the interface with the terminal.
- `LangEN_Poslynx.properties` - English language translation file, alternative translation maybe used by adding a new file and change the 2-character language code and translation text as appropriate.

---

---

**Note:** If the POS supports dynamic configuration, properties can be set there instead of in `poslynx.properties`.

---

---

## Core Classname

`manito.eft.poslynx.PoslynxCore`

This should be set as `EPSCore<x>` in `eftlinkconfig.properties`.

## Configuration Settings

The full set of configuration properties is defined and commented in `poslynx.properties`.

## Key Settings

Settings that may be different for each POS/PED.

The Core will connect to the PoslynxMINI device, this device is connected to the internet and the LAN and is accessible by Merchant Link. The device is DNS enabled and will have an address of the following format:

| Setting          | Description          | Example  |
|------------------|----------------------|--|
| Terminal address | IP of Mx925 terminal | TerminalIP = <b>xxxxxx</b> .poslynx.org<br><br>Where <b>xxxxxx</b> is the last 6 digits of the MAC address on the TL device. |

## Secondary Settings

| Setting          | Description  | Default | Example            |
|------------------|--|---------|--------------------|
| Terminal address | Port number  | 20100   | TerminalPort =     |
| Timeout          | Overall response timeout in seconds  | 120     | ResponseTimeout =  |
| EmbeddedPrinting | Option to buffer customer printout generated during the transaction and then include it in the POS authorisation response so that it can be merged with the POS receipt line to form a single receipt/voucher. | false   | EmbeddedPrinting = |

## Supported Functions

Below is a list of supported functionalities of the interface to Merchant Link. Many functionalities are provided by Merchant Link, such as Loyalty, Cashback and so on. (please refer to interface specification for details) but are not implemented because of the business requirement.

## Administration Functions

The terminal has some administration/maintenance functions. These are normally invoked from a dedicated "EFT Maintenance" button, but if this is not available, they could be accessed by an engineer using the EFTLink built-in test harness.

EFTLink uses DeviceProxy messages to display input prompts on the POS to manage these functions.

| Function              | Description  |
|-----------------------|--|
| Terminal Connect Test | Sends a request to test the connection to the device. This Core builds an XML request with the "CONNECTIONTEST" command and should receive a response with an "APPROVED" result if successful. |
| Reset Pin Pad         | Sends a request to reset the pin pad. This command will force the terminal to restart after acquiring Certificate Authority Public Key (CAPK) file from the host processor.                    |

|                          |  |
|--------------------------|--|
| Signature Capture        | Sends request to test the signature capture functionality on the device.   |
| Batch Summary            | Sends a batch summary request to the terminal. This provides a summary report of the current information for that batch of transactions prior to settlement. It should be considered as flash report typically referred to as an X read report.  |
| Batch Report             | Sends a batch card totals request to the terminal. This will return a report which will summarise the card type totals for the current batch prior to settlement.  |
| Batch Close              | Sends a batch close request to the terminal. This will initiate a batch close process on the host processor which triggers transactions to be submitted for financial settlement. This returns a day end report.   |
| Payment                  | <p>Sends payment request to the terminal. Terminal will return a response message with unformatted receipt strings for customer and/or merchant receipts.</p> <p>In an event of referral where authorization cannot be obtained online then a prompt for authorization code will appear; authorization code must be obtained via telephone and entered here.</p> <p>If successful, appropriate receipts will be printed at the end of transaction.</p> |
| Reversal                 | Sends reversal request to the terminal. This will reverse a transaction specified by the transaction number, found on the receipt, which must be captured by the POS and pass on to EFTLink.   |
| Refund                   | Sends refund request to the terminal. This will refund a transaction with specified amount.  |
| Sale State Notifications | Sends line items through to the device so the customer display can be updated in line with the POS.  |
| SVC Payment              | Sends a Gift or Merchandise credit card payment request to the terminal. If there are not enough funds available, only the funds available will be deducted. The POS client will have to settle the transaction with another tender in this scenario.  |
| SVC Activate             | Sends a Gift or Merchandise credit card activation request to the terminal.  |
| SVC Deactivate           | Sends a Gift or Merchandise credit card deactivation request to the terminal. The account is disabled after this as the request is intended to be used for lost or stolen cards. It is not possible to use the card or account once this request has been issued and accepted.   |
| SVC Add Value            | Sends a Gift or Merchandise credit card add value request to the terminal. This will only add value to an account that has been activated.   |
| SVC Balance Enquiry      | Sends a Gift or Merchandise credit card balance enquiry request to the terminal.   |

|                      |  |
|----------------------|--|
| SVC Unload (Cashout) | Sends a Gift or Merchandise credit card cash out request to the terminal. All funds are deducted from the account and the cash back amount is returned to the POS. The account is not deactivated as part of this process. |
|----------------------|--|

## General Information

This section of the document covers EFTLink Integration with SixPay Payment Systems. It should be read in conjunction with the *Oracle Retail EFTLink Framework Installation and Configuration Guide*.

## EFTLink General

This document assumes static EFTLink configuration. When deploying with a POS that supports dynamic configuration, all property settings referred to below should be set on the POS, and not directly into local property files.

## System Architecture

Six Payment Services MPD is deployed as a store server application to manage the connection to the authorization host and to handle all the local PEDs. PEDs use IP, so must be connected to the LAN. EFTLink connects to the store server, not directly to any PED. EFTLink communicates with MPD using an implementation of the IFSF/OPI protocol.

---



---

**Note:** This document does not cover the installation of MPD.

---



---

## Fileset

In addition to standard EFTLink files the following are used:

- Cores/SixPay/sixpaycore.jar – executable code for the MPD OPI interface
- sixpay.properties – configuration settings to specify which features are enabled and to define communication parameters for the interface with the store server.

---



---

**Note:** If the POS supports dynamic configuration, properties can be set there instead of in `sixpay.properties`.

---



---

## Language

There are translation files in `sixpaycore.jar`, that should not need to be modified, but if a translation needs to be changed, they can be extracted to the base `eftlink` folder.

```
LangDE_Sixpay.properties
LangEN_Sixpay.properties
LangFR_Sixpay.properties
LangIT_Sixpay.properties
LangNL_Sixpay.properties
```

The language used will follow the language set in the EFTLink framework; see the *Oracle Retail EFTLink Framework Installation and Configuration Guide*, EFTLink General Information, Translation section

```
EftlinkConfig.properties
DisplayLanguage = EN
```

Possible values for SixPay are: DE, EN, FR, IT, NL



## Core Classname

This should have been set as `EPSCore0=manito.eft.sixpay.SixpayMPDOPIClient` in `eftlinkconfig.properties` by `installcore.bat` or `installcore.sh`

## Configuration Settings

Configuration settings are made in `sixpay.properties`, which would have been copied from `cores/SixPay` to the base `eftlink` folder by `installcore.bat` or `installcore.sh`

### Key Configuration Settings

| Setting                     | Description  | Default   | Example   |
|-----------------------------|--|-----------|---|
| SixpayServerIP              | IP address of the store server running MPD   | 127.0.0.1 | SixpayServerIP = 10.0.0.50  |
| SixpayWorkstationID         | Optional Setting for specific WorkstationID, and to set the WorkstationID format.<br><b>Note:</b> This becomes the base number when SixpayWorkstationIDPosBased is enabled. The default is for this not to be set (property is commented) - the workstation number will be taken directly from the OPI message from the POS. |           | SixpayWorkstationID = POS1  |
| SixpayWorkstationIDPosBased | Option to automatically set the MPD workstation ID from the numeric suffix of a mixed numeric/ non-numeric POS workstation ID. Boolean.<br>If this feature is enabled, the SixpayWorkstationID setting is taken as the value for POS #1 and the numerical component is incremented for all other POSs.                       | false     | WorkstationIDPosBased = true<br>This would mean that for POS2 with the SixpayWorkstationID = POS1 set above, messages to MPD would be from POS2.<br>Careful use of WorkstationID settings and overrides in both the POS and EFTLink should make it possible to deploy a standard <code>sixpay.properties</code> file across all POSs. |

### Optional Configuration Settings

These settings are normally left on defaults.

| Setting        | Description                                 | Default | Example                |
|----------------|---|---------|------------------------|
| SixpayChannel0 | TCP/IP port used for primary channel to MPD | 20002   | SixpayChannel0 = 20002 |
| SixpayChannel1 | TCP/IP port for device requests from MPD    |         | SixpayChannel1 = 20007 |

| Setting               | Description  | Default    | Example                     |
|-----------------------|--|------------|-----------------------------|
| SixpayResponseTimeout | Timeout in seconds for EFTLink to wait for the response from MDP.  | 300        | SixpayResponseTimeout = 300 |
| IncludeSaleItems      | If enabled, sale item details are included in the payment request.   | false      | IncludeSaleItems = true     |
| EmbeddedPrinting      | Whether customer printout is to be buffered and included in the POS authorization response such that it can be embedded in the POS receipt.                            | false      | EmbeddedPrinting = false    |
| ElectronicJournal     | Whether merchant printout (other than signature slips) is buffered and included in the POS authorization response such that it can be stored in an electronic journal. | false      | ElectronicJournal = false   |
| SignatureCheckTag     | Trigger tag/text to detect that a signature has been asked for and should be checked, default "Signature:"   |            | SignatureCheckTag=sign      |
| SignatureCheckTimeout | Timeout for Signature OK? Question.  | 30 seconds | SignatureCheckTimeout = 30  |

## Fixed Configuration Settings

The property file `sixpay.properties` has a section of settings headed as Fixed Configuration settings, which should not be changed.

## Other Information

### PED Identification/Selection

The PED is identified to MPD by the `WorkstationID` in the IFSF/OPI message. By default, this is copied through from the `WorkstationID` in the POS-EFTLink message. Thus, the POS numbering needs to be kept in sync with the PED configuration in MPD. If this is not possible, or if the POS uses non-numeric `WorkstationID`, override settings must be used in the `sixpay.properties` files as described above.



---

---

# SolveConnect Core

## General Information

This document covers EFTLink Integration with TLG (The Logic Group) Payment Systems. It should be read in conjunction with the *Oracle Retail EFTLink Framework Installation and Configuration Guide*.

## EFTLink General

This document assumes static EFTLink configuration. When deploying with a POS that supports dynamic configuration, all property settings referred to below should be set on the POS, and not directly into local property files.

## System Architecture

EFTLink connects directly to the SolveConnect software usually installed on the same PC as the POS, using a proprietary socket protocol.

---

---

**Note:** This document does not cover the installation of SolveConnect software.

---

---

## Fileset

In addition to standard EFTLink files the following are used:

- Core/SolveConnect/SolveConnect.jar – Core interface to TLG's SolveConnect software.
- SolveConnect.POS.properties
- SolveConnect.properties

## Language

There are translation files in SolveConnect.jar, that should not need to be modified, but if a translation needs to be changed, they can be extracted to the base eftlink folder.

```
LangEN_SolveConnect.properties  
LangES_SolveConnect.properties
```

The language used will follow the language set in the EFTLink framework; see the *Oracle Retail EFTLink Framework Installation and Configuration Guide*, EFTLink General Information, Translation section

```
EftlinkConfig.properties
```

```
DisplayLanguage = EN
```

Possible values are: EN, ES

## Core Classname

This should have been set as EPSCore0= manito.eft.solveconnect.SolveConnectCore in EftlinkConfig.properties. by installcore.bat or installcore.sh

## Configuration Settings

There are two configuration files – `SolveConnectPOS.properties` and `SolveConnect.properties`. These are copied from `cores/SolveConnect` to the base `eftlink` folder by `installcore.bat` or `installcore.sh`.

`SolveConnectPOS.properties` carries only the POS specific identifiers, `SolveConnect.properties` carries everything else, and can usually be deployed on a retailer's estate without other changes.

## Key Settings

`SolveConnectPOS.properties`

| Setting  | Description   | Default | Example             |
|----------|---|---------|---------------------|
| SourceID | The POS specific identifier, allocated by retailer, to be unique across the retailer's estate.    |         | SourceID = DPOS0001 |
| Store.ID | A 4-digit store identifier which forms part of the reference number assigned to each transaction. | 9999    | Store.ID = 1234     |
| POS.ID   | 2 digit POS identifier which forms part of the reference number assigned to each transaction.     | 99      | POS.ID = 25         |

**Note:** Together, the Store.ID and POS.ID settings can be used to create a transaction reference that will be unique across all sites in a group.

## Secondary Configuration Settings

`SolveConnect.properties`

| Setting                    | Description  | Default | Example                        |
|----------------------------|--|---------|--------------------------------|
| ServiceHost                | Hostname or IP address of SolveConnect service.  |         | ServiceHost = 127.0.0.1        |
| TransactionTimeout Period  | Number of seconds to allow a transaction to complete.  | 180     | TransactionTimeoutPeriod = 180 |
| CancellationTimeout period | Maximum number of seconds the core will wait for a transaction response following a cancellation.  | 30      | CancellationTimeoutPeriod = 30 |
| MaintenanceMenuTimeout     | The number of seconds to wait for an option to be selected before dismissing the Maintenance menu. | 30      | MaintenanceMenuTimeout = 45    |
| AuditLoggingEnabled        | Enable/Disable logging of transaction results to an audit log.                                     | false   | AuditLoggingEnabled = false    |

| Setting                    | Description   | Default     | Example   |
|----------------------------|---|-------------|---|
| TransactionReferenceScheme | The format and source of Store and Till-ID values. Recognised values are Properties and PowerPOS.<br><br>If set to PowerPOS, the POS.ID value will be automatically extracted from the POS system name at run time, so the setting in <code>SolveConnect.POS.properties</code> can be left at zero. | Properties  | <code>TransactionReferenceScheme = PowerPOS</code>  |
| TransactionNumberFromPOS   | Whether to use the transaction number from the POS (with suffixes to ensure uniqueness) rather than the default auto-incrementing number.   | true        | <code>TransactionNumberFromPOS = true</code>  |
| ForcePurchaseWithCashback  | Force all POS Purchase requests to be converted to Solve Purchase with Cashback requests.   | true        | <code>ForcePurchaseWithCashback = true</code>   |
| PromptForCashbackCharge    | Prompt for a cashback charge.   | true        | <code>PromptForCashbackCharge = true</code>   |
| TransactionReferenceFormat | Format for the transaction reference to be passed to SolveConnect. Built from the store id (S), POS ID (P) and POS Transaction number (T).  | SSSSPPTTTTT | <code>TransactionReferenceFormat = SSSSPPTTTTT</code>   |
| EmbeddedPrinting           | Whether customer printout is to be buffered and included in the POS authorization response such that it can be embedded in the POS receipt.   | false       | <code>EmbeddedPrinting = false</code>   |
| DCC Keywords               | DCC keywords for extracting DCC from status message. There are no defaults.   |             | <code>DCCAmountKeyword = DCC Amount</code><br><code>DCCExchangeRateKeyword = Exchange Rate</code><br><code>DCCMarginKeyword = Margin</code> |
| AuthTokenOrigin            | Whether to automatic token recognition to establish local/central origin.   | false       | <code>AuthTokenOrigin = false</code>  |
| Token Formats              | Token formats to identify local/central token. There are no defaults.   |             | <code>LocalTokenFormat = 1234567890123456789</code><br><code>CentralTokenFormat = 123456ABCDEFGH1234</code>                                 |
| CardSwipeTimeoutPeriod     | Number of seconds to allow for a standalone card read/swipe to complete.<br><br>This will need to be extended for example, to 9999 if an open/background card read operation is required.   | 30          | <code>CardSwipeTimeoutPeriod = 30</code>  |

| Setting                                 | Description   | Default | Example  |
|---|---|---------|--|
| PEDLogoffDelayTime                      | Delay time between POS logoff and PED logoff, to allow for operator changeover without PED disconnection. Applies to networked PEDs only. Time in seconds. Set to 0 to disable PED logoff.  | 300     | PEDLogoffDelayTime = 300                       |
| SelectiveMerchantPrint                  | Determine whether merchant print is selective. That is, enable for some conditions, disabled for others.  | false   | SelectiveMerchantPrint = false                 |
| MerchantPrint.not_present.not_performed | <p>In selective mode, all merchant print is disabled by default, but can be selectively re-enabled based on a combination of the transaction attributes returned by SolveConnect.</p> <p>Note - this is the opposite way round to selective customer print.</p> <p>The attributes used are:</p> <ul style="list-style-type: none"> <li>TRANSACTION:customer present, not_present, internet</li> <li>CARDHOLDER_RESULT:verification pin, signature, pin_and_signature, on_device, not_performed, failed, unknown</li> </ul> <p>These attributes are formed into a dot-separated property name (for example MerchantPrint.present.pin) that can be set to "true" to reenable merchant print for that attribute combination.</p> <p>Note - merchant print requiring signature will always be printed, it cannot be disabled for example, to re-enable merchant print for CustomerNotPresent transactions:</p> <p>MerchantPrint.not_present.not_performed = true</p> <p>Re-enable merchant print for CustomerNotPresent transactions.</p> |         | MerchantPrint.not_present.not_performed = true |
| SelectiveCustomerPrint                  | Determine whether customer print is selective that is, enabled for some conditions, disabled for others.  | false   | SelectiveCustomerPrint = false                 |

| Setting                                 | Description   | Default | Example   |
|---|---|---------|---|
| CustomerPrint.not_present.not_performed | <p>In selective mode, customer print is enabled by default, but it can be selectively disabled based on a combination of the transaction attributes returned by SolveConnect.</p> <p>Note - this is the opposite way round to selective merchant print.</p> <p>The attributes used are:</p> <ul style="list-style-type: none"> <li>TRANSACTION:customer present, not_present, internet</li> <li>CARDHOLDER_RESULT:verification pin, signature, pin_and_signature, on_device, not_performed, failed, unknown</li> </ul> <p>These attributes are formed into a dot-separated property name for example, (CustomerPrint.present.pin) that can be set to "false" to disable customer print for that attribute combination.</p> <p>For example, to disable customer print for CustomerNotPresent transactions:</p> <pre>CustomerPrint.not_present.not_performed = true</pre> |         | <pre>CustomerPrint.not_present.not_performed = true</pre> |
| SignatureCheckReprintOption             | <p>Determine whether to include a "reprint" option when prompting operator for signature verification. Default false.</p> <p>Caution - if set true, the display request will be sent as a menu selection rather than a yes/no and this will affect the way it is presented to the operator.</p>   |         | <pre>SignatureCheckReprintOption = true</pre>             |
| ManualAuthMinLength                     | <p>Minimum input length required for Manual/Voice referral authorization code response.</p>   | 0       | <pre>ManualAuthMinLength = 0</pre>                        |

## Fixed Configuration Settings

There are a number of fixed configuration settings in `SolveConnect.properties` that are commented in the property file. These are advanced options for development use.



## Supported Functions

The following operations are supported by this implementation of the SolveConnect interface.

| Function        | Description   |
|-----------------|---|
| Payment         | <p>Sends payment request to the terminal. Terminal will return a response message with formatted receipt strings for customer and/or merchant receipts.</p> <p>In an event of referral where authorization cannot be obtained online then a prompt for authorization code will appear; authorization code must be obtained via telephone and entered here. If successful, appropriate receipts will be printed at the end of transaction.</p> |
| Reversal        | <p>Sends reversal request to the terminal. This will reverse a transaction specified by the transaction number, found on the receipt, which must be captured by the POS and passed on to EFTLink.</p>   |
| Refund          | <p>Sends refund request to the terminal. This will refund a transaction with specified amount.</p>  |
| GiftCard        | <p>Sends giftcard payment request to the terminal. Specified amount will be deducted from the giftcard.</p> <p>Administration options to add balance and check balance is also supported.</p>   |
| Receipt Reprint | <p>Reprint merchant/customer receipt.</p>   |

## EFTLink General

This document assumes static EFTLink configuration. When deploying with a POS that supports dynamic configuration, all property settings referred to below should be set on the POS, and not directly into local property files. It should be read in conjunction with the *Oracle Retail EFTLink Framework Installation and Configuration Guide*.

## Minimum Version

The Transax interface requires a minimum EFTLink version of v15.0.

## System Architecture

EFTLink connects to FIS TransaxEFT software running on the same PC as the POS via TCP/IP sockets as an OPI client.

---

---

**Note:** This document does not cover installation of FIS TransaxEFT software.

---

---

## Fileset

In addition to standard EFTLink files, TransaxEFT uses:

- `cores/TransaxEFT/transaxeftcore.jar` - executable code for the TransaxEFT core.
- `TransaxEFT.properties` - configuration settings to specify which features are enabled and to define communication parameters for the interface with the EFT payment system.

## Translation

There is an English translation file in `transaxeftcore.jar`, that should not need to be modified, but if a translation needs to be changed, it can be extracted to the base `eftlink` folder.

`LangEN_TransaxEFT.properties`

The file in use follows the language setting for EFTLink itself, defined in `EftlinkConfig.properties`, so the only possible setting is EN; the default.

Example

```
DisplayLanguage = EN
```

## Core Classname

The following should have been set in the `EftlinkConfig.properties` file by `installcore.bat` or `installcore.sh`

```
EPSCore0 = manito.eft.transaxeft.TransaxEFTOPIClient
```

## Configuration Settings

Settings are defined in `TransaxEFT.properties`.

### Key Settings

These are no settings that must be set differently per POS.

### Secondary Settings

These settings should not need adjustment, but are defined here.

| Setting                             | Description  | Default                              | Example   |
|-------------------------------------|--|--------------------------------------|---|
| TransaxEFTChannel0                  | The TCP port on which the Core sends requests and device responses to TransaxEFT.  | 8900                                 | TransaxEFTChannel0 = 8900                                 |
| TransaxEFTChannel1                  | The TCP port on which the Core listens for responses and device requests from TransaxEFT.  | 9900                                 | TransaxEFTChannel1 = 9900                                 |
| ReceiptFormatFile                   | Name and path of the receipt XSLT translation file used to format the receipt text.  | transaxeft\\AccreditationReceipt.xml | ReceiptFormatFile = transaxeft\\AccreditationReceipt.xml  |
| BalanceEnquiryFormatFile            | Name and path of the card balance enquiry XSLT translation file.   | transaxeft\\Balance.xml              | BalanceEnquiryFormatFile = transaxeft\\Balance.xml        |
| ReconciliationFormatFile            | Name and path of the reconciliation report XSLT translation file.  | transaxeft\\Reconciliation.xml       | ReconciliationFormatFile = transaxeft\\Reconciliation.xml |
| ReceiptTextPassThroughEnabled       | Only enable when TransaxEFT provides pre-formatted plain text receipt lines rather than name/value pairs.  | false                                | ReceiptTextPassThroughEnabled = false                     |
| TransaxEFTResponseTimeout           | Extend the period we will wait for a CardServiceResponse message after sending our request.  | 120                                  | TransaxEFTResponseTimeout = 120                           |
| TransaxEFTOperatorRecoverySupported | Do not allow the operator to specify the success or failure of a transaction.  | false                                | TransaxEFTOperatorRecoverySupported = false               |
| MaintenanceTimeout                  | Specifies the maximum number of seconds to wait for the operator to select an administration menu option. If no option is selected the maintenance function is completed and control passes back to the POS. | 60                                   | MaintenanceTimeout = 60                                   |
| MaintMenuOptReceiptReprintEnabled   | Enable/disable the maintenance menu option TXT_REPRINT. Set to false if, besides the maintenance menu, the POS has its own means of requesting a ticket reprint. Set to true for Power POS.                  | true                                 | MaintMenuOptReceiptReprintEnabled = false                 |

| Setting                            | Description   | Default | Example                                   |
|------------------------------------|---|---------|---|
| MaintMenuOptEODEnabled             | Enable/disable the TXT_RECONCILIATION_WITH_CLOSURE menu option. Set to false if the POS has its own means of requesting reconciliation with closure.  | true    | MaintMenuOptEODEnabled = true             |
| MaintMenuOptPEDTestEnabled         | Enable/disable the menu option TXT_PED_TEST. Set to false for Power POS as it cannot display multiple lines of text received in cashier display and cashier input device requests.                  | false   | MaintMenuOptPEDTestEnabled = false        |
| MaintMenuOptPEDTestPrintedEnabled  | Enable/disable the menu option TXT_PED_TEST_PRINTER_OUTPUT. Set to true for Power POS.  | true    | MaintMenuOptPEDTestPrintedEnabled = true  |
| MaintMenuOptEODQueryEnabled        | Enable/disable the menu option TXT_QUERY_LAST_RECONCILIATION. Set to false for Power POS as it cannot display multiple lines of text received in cashier display and cashier input device requests. | false   | MaintMenuOptEODQueryEnabled = false       |
| MaintMenuOptEODQueryPrintedEnabled | Enable/disable the menu option TXT_QUERY_LAST_RECONCILIATION_PRINTER_OUTPUT.  | true    | MaintMenuOptEODQueryPrintedEnabled = true |

## Fixed Settings

There are a small number of settings included in `TransaxEFT.properties` that should not be changed. All settings not documented here fall into that category.

## Supported Functions

Below is a list of supported functionalities of the interface to Transax.

| Function                    | Description   |
|-----------------------------|---|
| Payment                     | EFTLink sends payment requests to Transax. Transax will return a response message with formatted receipt strings for customer and/or merchant receipts.<br>If successful, appropriate receipts will be printed at the end of transaction. |
| Refund                      | EFTLink sends refund requests to Transax. This will refund a transaction with specified amount.   |
| Reconciliation / Settlement | Prints a settlement report at day end.  |
| Receipt Reprint             | Reprint merchant/customer receipt.  |
| Loyalty Balance Enquiry     | Check the balance of a loyalty card.  |

## Maintenance Options

The Transax interface has some administration/maintenance operations. These are normally invoked from a dedicated “EFT Maintenance” button at the POS, though the content of the display screen then presented to the operator is controlled by MaintMenu options enabled above.

- Reprint last receipts
- Reconciliation with Closure (EFT EOD)
- Test PED (Display results)
- Test PED (Print results)
- Test connection to Authorisation Host
- Test printer
- Query reconciliation figures (Display results)
- Query reconciliation figures (Print results)

---

---

# VeriFone Ocius Sentinel

## General Information

### Overview

This document covers EFTLink Integration with Ocius Sentinel Payment Systems. It should be read in conjunction with the *Oracle Retail EFTLink Framework Installation and Configuration Guide*.

### System Architecture

EFTLink connects to the Ocius Sentinel application using a proprietary socket protocol. Normally the Ocius Sentinel application, which is configured to run screenlessly, is installed on the same PC as the POS application.

---

---

**Note:** This document does not cover the installation of the Ocius Sentinel application itself.

---

---

### Fileset

In addition to standard EFTLink files:

- Cores/OciusSentinel/ociussentinelcore.jar
- ocius.properties
- ocius\_receipt.properties (only if using XML receipt data, can be auto-deployed, see XML Receipts).
- receipt template files (only if using XML receipt data, can be auto-deployed , see XML receipts).

### Language

There are no translation files in ociussentinelcore.jar

Ocius Sentinel is deployed in the UK, so the language set in the EFTLink framework should be English, which is the default.

See the *Oracle Retail EFTLink Framework Installation and Configuration Guide*, EFTLink General Information section, Translation sub-section.

```
EftlinkConfig.properties
```

```
DisplayLanguage = EN
```

### Core Classname

The following should have been set in `eftlinkconfig.properties` by `installcore.bat` or `installcore.sh`

```
EPSCore0 = manito.eft.ocius_sentinel.OciusSentinelCore
```

## Configuration Settings

The core is configured via properties contained in the `ocius.properties` file, which is copied from `cores/OciusSentinel` folder to the root `eftlink` folder by `installcore.bat` or `installcore.sh`.

### Key Settings

These must be set. Since these two properties must be encrypted by default, see [Password Encryption](#).

| Setting               | Description  | Example  |
|-----------------------|--|--|
| <code>user.id</code>  | The user ID to send to the terminal when logging on. The ID is allocated by the Ocius Sentinel, and needs to be encrypted for default configuration.   | <code>user.id=89eb96f2dfed02384e99fb7f8bfea610</code>  |
| <code>user.pin</code> | The user PIN to send to the terminal when logging on. The PIN is allocated by the Ocius Sentinel, and needs to be encrypted for default configuration. | <code>user.pin=89eb96f2dfed02384e99fb7f8bfea610</code> |

### Optional Configuration Settings

There are a large number of optional settings that usually do not need to be set or modified, but for completeness they are defined here. In the property file all are commented with default values or empty.

| Setting                                  | Description   | Default   |
|--|---|-----------|
| <code>ip.address</code>                  | The IP address of the Ocius Sentinel software. The default is 127.0.0.1, which will work as long as the Ocius Sentinel software is installed on the POS PC.   | 127.0.0.1 |
| <code>ip.port</code>                     | The IP port of the terminal.  | 25000     |
| <code>terminal.menu.configuration</code> | The menu configuration to send to the terminal when logging on. The default is * which enables all menus. See the Ocius Manual for more details.  | *         |
| <code>account.id</code>                  | The account ID to send with each transaction. This option is used in some deployments, and Verifone would indicate the value to use.  | blank     |
| <code>auto.logon</code>                  | If this is set true then the core will log on to the terminal automatically when it receives a transaction (if the POS has not already sent a logon command).   | true      |
| <code>pause.before.auto.logon</code>     | The number of milliseconds to wait before issuing an automatic logon command to Sentinel. This is to allow for an issue with Sentinel which causes it to occasionally reject or lose messages which are sent too soon after a previous communication. | 1000      |
| <code>auto.logon.pause</code>            | The number of milliseconds to wait after an auto logon before sending a transaction. The pause should be for several seconds.   |           |

| Setting                        | Description  | Default  |
|--------------------------------|--|--|
| merchant.receipt.path          | The folder where Ocius Sentinel is to place the merchant receipt. If undefined (commented or blank value) the file would be expected at the root of the same drive, which is where Ocius Sentinel puts the receipt by default.                               |  |
| merchant.receipt.filename      | The name that Ocius Sentinel will use for the merchant receipt. Default is Receipt1.txt, it can be modified in the Ocius Sentinel application, and if so the name used should be entered here.   | Receipt1.txt   |
| customer.receipt.path          | The folder where Ocius Sentinel is to place the customer receipt. This is only relevant if xml. If undefined (commented or blank value) the file would be expected at the root of the same drive, which is where Ocius Sentinel puts the receipt by default. |  |
| customer.receipt.filename      | The name that Ocius Sentinel is to use for the customer receipt. Default is Receipt2.txt. This can be modified in the Ocius Sentinel application, and if so, the name used must be entered here.   | Receipt2.txt   |
| report.path                    | The folder where Ocius Sentinel is to place the report file.   |  |
| report.filename                | The name that Ocius Sentinel is to use for the report file.  |  |
| progress.ip.port               | The port that the core listens on for status messages from Ocius Sentinel.   | 25001  |
| tear.merchant.receipt.text     | The text to be displayed at the POS when prompting the operator to remove the merchant receipt from the printer.   |  |
| tear.customer.receipt.text     | The text to be displayed at the POS when prompting the operator to remove the customer receipt from the printer.   |  |
| strip.receipt.carriage.returns | Ocius Sentinel delivers receipts with lines terminated by both carriage return and linefeed characters. If this option is set true then the carriage return characters will be removed.  | false  |
| max.cashback.length            | The maximum length permitted for a cashback amount.  | 5  |
| duplicate.receipt.title        | An extra title to add to the top of a receipt which is reprinted in response to the "Re-print/Continue" message.   | *** Duplicate Receipt<br>***\n<br>where the \n indicates a linefeed. Leave blank to suppress this title. |
| suppress.merchant.receipt      | Whether to suppress printing of the merchant receipt so only a customer copy is provided.  | false  |
| offer.reprint                  | Whether to display the "Re-print/Continue" dialogue after printing a receipt.  | true   |



| Setting                  | Description  | Default  |
|--------------------------|--|--|
| defer.customer.receipt   | If true this will cause the customer receipt to be sent as part of the final CardServiceResponse when payment processing is complete.  | false  |
| account.on.file.mode     | This may be set to an integer from 0 to 4 inclusive. Values are defined in the Ocius Sentinel integration guide v1.5 as follows:<br>0 - Not Set<br>1 - Do Not Register (the default)<br>2 - Register<br>3 - Register Only<br>4 - Register, decline transaction if registration fails.  |  |
| card.read.mode           | This may be set to 0, 1 or 2 and defines what type of card is to be read when the core receives a card read request:<br>0 - Non EFT card<br>1 - EFT card<br>2 - Automatic based on the EFTLink background flag set by the POS, background=true reads a non-EFT card, otherwise an EFT card is expected (this is the default behaviour for this setting). |  |
| remove.card.after.read   | If true this should cause Ocius Sentinel to prompt for the card to be removed after a card read. In practice it has been found that Sentinel ignores this setting.   |  |
| encrypted.passwords      | user.id, user.pin, account.id and transax.account.id must be encrypted using the encryption utility see <a href="#">Password Encryption</a> .  |  |
| auto.confirm.licence.key | If true (the default), then there will be an automatic response to the LicenceDetailConfirmation status from Ocius Sentinel.   | true   |
| card.wait.mode           | If true the core will send CARDWAIT records, otherwise it will operate in standard mode.   | false  |
| wait.record.header       | This is the header text to display on the PED when it prompts for the card details to be presented.  | The default is for the section to be left blank. |
| wait.record.body         | This is the body text to display on the PED when it prompts for the card details to be presented.  | The default is for the section to be left blank. |
| wait.record.footer       | This is the footer text to display on the PED when it prompts for the card details to be presented.  | The default is for the section to be left blank. |
| wait.record.timeout      | This is the time in seconds for the PED to wait for the card details to be presented.  | 0 (no timeout)                                   |

| Setting                      | Description   | Default  |
|------------------------------|---|--|
| wait.record.capture.methods  | <p>This is a hex bitmap of the capture methods that the PED is to allow.</p> <p>The hex bitmap is comprised of the following hex values:</p> <p>Keyed = 01<br/> Swipe = 02<br/> ICC = 04<br/> Reserved = 08</p>   | <p>The default is for the core to leave this blank, in which case Sentinel will apply the following default:</p> <p>ICC + Swipe + Keyed = 07</p>                                     |
| wait.record.fallback.methods | <p>This is a hex bitmap of the fallback methods that the PED is to allow.</p> <p>The hex bitmap is comprised of the following hex values:</p> <p>Fallback from ICC to Swipe = 01<br/> Fallback from Swipe to Key = 02</p>   | <p>The default is for the core to leave this blank, in which case Sentinel will apply the following default:</p> <p>Fallback from ICC to Swipe + Fallback from Swipe to Key = 03</p> |
| auto.offline                 | If true the core will automatically instruct Ocius Sentinel to work offline if the remote server is unavailable.  | false  |
| reference                    | <p>This setting configures the customer reference generated by the core. It may contain any text except commas but the following case-sensitive keywords will be substituted with corresponding data:</p> <p>Date: the transaction date provided by the POS in the form YYMMDD<br/> Time: the transaction time provided by the POS in the form HHMMSS<br/> Transnum: the transaction number provided by the POS<br/> User: the operator ID provided by the POS when it logged on to EFTLink<br/> Pos: the POS ID provided by the POS when it logged on to EFTLink</p> | date transnum user pos   |
| CARDWAIT with CNP            | <p>If a card swipe request is issued with the CNP flag set then an alternative set of wait record parameters will be sent to the PED. These have the same names as the wait record properties already defined but with .cnp appended, for example:</p> <p>wait.record.capture.methods.cnp = 1</p> <p>The primary purpose of this is to allow the PED to be forced into keyed only mode in a customer-not-present (telesales) scenario. The definitions and default settings for the alternative parameters are the same as the standard parameters.</p>               |  |
| simple.cnp.enabled           | <p>For telesales if a card has been keyed via a previous card swipe and customer address capture is not required as part of the subsequent transaction then this setting should be set true.</p> <p><b>Note:</b> In this mode &lt;CNP&gt;true&lt;/CNP&gt; is added to the XML receipt data for telesales.</p>   | false  |
| transax.account.id           | The account ID to use for Transax transactions.   |  |

| Setting                               | Description  | Default |
|---------------------------------------|--|---------|
| transax.types.requiring.card          | The Transax transaction types which require card entry at the PED. This may be any combination of the letters A, B, M, O or P without spaces or separators.  | P       |
| transax.declined.operator.message     | If a Transax payment is declined or otherwise fails this optional setting can be used to provide an acknowledgeable message to bring the failure to the attention of the operator. The default value is blank (no message will be displayed). If required the value may be static text. For example:<br>transax.declined.operator.message=Transax Payment Void<br>or it may be used to display one of the fields of a Transax XML receipt. For example:<br>transax.declined.operator.message=<Message> |         |
| auto.confirm.auth.code                | If this is set true then Ocius Sentinel status 20 (Confirm Auth Code) will be answered automatically.  |         |
| voice.referral.amount.text            | This defines the label shown against the transaction amount in the voice referral prompt. If the POS already displays the amount elsewhere on the screen then voice.referral.amount.text may be set to blank to exclude it from the message sent by the core.  |         |
| voice.referral.compact.dialogue       | If true the two stage referral dialogue where the operator must first confirm that the authorization has been accepted before entering the authorization code will be reduced to a single dialogue where the operator may immediately enter an authorization code or blank to cancel.  |         |
| signature.verification.reprint.option | By default the signature verification dialogue offers two options to confirm or reject the signature. If this setting has a value a third option will be displayed which will cause the signature slip to be reprinted. The value should be the text to be displayed, for example Reprint. The default is blank which disables this option.<br><b>Note:</b> offer.reprint provides a more general purpose reprint mechanism.   |         |
| defer.void.receipts                   | If true then void customer receipts will not be printed immediately but will be embedded in the final response from the core. Applies only in XML mode.  | false   |
| suppress.final.declined.message       | If the POS displays its own declined message on receiving a payment failure response from the core then this setting may be used to suppress any similar display message from the core.  | false   |
| suppress.cnp.signature.receipt        | If true then the signature receipt will be suppressed for telesales transactions when simple.cnp.enabled is true. Applies only for XML based receipts.   | true    |
| auto.translate.status.messages        | Indicates whether the core should translate status messages according to the recommendations in the Ocius Sentinel Integration Guide. If false then status messages can still be translated.   | false   |

| Setting                       | Description  | Default   |
|-------------------------------|--|---|
| space.out.status.messages     | Indicates whether status text from Ocius Sentinel should be spaced out for display, for example <code>ExpiryDateRequired</code> becomes <code>Expiry Date Required</code> .  | true  |
| ped.unavailable.retry.pause   | If status message 55 (PEDUnavailable) is received this setting specifies the number of milliseconds to wait before requesting Ocius Sentinel to retry.   | The default is 0 (zero) which disables handling of status message 55. |
| legacy.printing               | Enables file-based printing if set to true, otherwise socket-based printing will be used.  | true  |
| cancel.card.wait.delay        | When <code>card.wait.mode=true</code> this setting defines the minimum interval in milliseconds between a card swipe request from the POS and a cancellation of the card swipe (abort). This is to allow for a limitation in Ocius Sentinel which cannot cope with the two messages being sent in close proximity. The delay is only applied if needed and the default interval is 1000ms. | 1000ms  |
| max.login.ready.wait          | After a processing a login request from the POS this is the maximum time to wait in milliseconds for a Ready status from Ocius Sentinel before returning a login success response to the POS. If this setting is zero then the wait will be indefinite.  | zero  |
| await.ready.after.transaction | The default behavior for the core is to wait for Ocius Sentinel to complete all necessary actions after a payment including having the customer remove the card from the PED before responding to the POS with the result. To allow the transaction to complete at the POS without waiting for card removal set <code>await.ready.after.transaction=false</code> .                         |   |
| store.merchant.receipt        | If true the merchant receipt will not be printed but will be sent to the POS to be stored in an electronic audit journal (where the POS supports this capability).   | false   |
| use.ocius.card.text           | If true EFTLink will use the card scheme name provided by Ocius Sentinel rather than performing a look-up in its Card Range File.  | false   |
| separate.receipt.lines        | If true the deferred (embedded) customer receipt will be sent as separate lines rather than as a single block of text containing line breaks. This is to cater for POS systems which have a limit to the length of continuous text that they can accept.   | false   |
| auto.logoff                   | If the response to a logon request to Ocius Sentinel indicates that a user is already logged in then this setting will cause the core to send a logoff followed by another logon.  | false   |
| deploy.default.templates      | If true then a default set of receipt templates will be created by EFTLink if they do not already exist in the EFTLink folder at start up. Applies only when XML receipts are in use.  | false   |
| dummy.void.receipts           | If true then the core will generate a dummy success response and receipt for a payment refund request without any interaction with Ocius Sentinel.   | false   |
| fixed.receipt.merchant.text   | When using Ocius Sentinel's preformatted receipts (as opposed to XML based receipts) this defines the text within the receipt which identifies it as a merchant receipt.   | MERCHANT COPY   |

| Setting                           | Description   | Default  |
|-----------------------------------|---|--|
| fixed.receipt.custome<br>r.text   | When using Ocius Sentinel's preformatted receipts (as opposed to XML based receipts) this defines the text within the receipt which identifies it as a customer receipt.  | CARDHOLDER COPY  |
| fixed.receipt.signatur<br>e.text  | When using Ocius Sentinel's preformatted receipts (as opposed to XML based receipts) this defines the text within the receipt which identifies it as a signature receipt.   | Please Sign Below.   |
| fixed.receipt.void.tex<br>t       | When using Ocius Sentinel's preformatted receipts (as opposed to XML based receipts) this defines the text within the receipt which identifies it as a void receipt.  | VOID   |
| fixed.receipt.decline<br>d.text   | When using Ocius Sentinel's preformatted receipts (as opposed to XML based receipts) this defines the text within the receipt which identifies it as a declined receipt.  | DECLINED   |
| download.retry.limit              | As part of the login process Ocius Sentinel may detect and attempt to download a software update. It is possible at this stage for Sentinel to send status 75 (Download Still Being Prepared) in which case this setting defines the number of times to retry the software download.  | 1 which indicates unlimited retries.                               |
| cancel.download.on.f<br>ailure    | If a software download fails due to reaching the retry limit, this setting defines whether a download cancellation command should be sent to Ocius Sentinel in order to allow the POS to login and proceed with sales operations. If no cancellation command is sent then the operator will need to interact with the (Windows) Ocius Sentinel application manually in order to cancel the download or attempt further retries. | true   |
| ocius.sentinel.exe.pat<br>h       | After a successful software download Ocius Sentinel will send status 58 (Restart After Software Update) indicating that it needs to be restarted. In response to this the core will send a message instructing Ocius Sentinel to shut down and will then re-launch the application by running an executable file, the location of which is defined by this setting.   | \Program<br>Files\Verifone\Ocius<br>Sentinel\OciusSentinel<br>.exe |
| ocius.sentinel.restart.<br>pause  | When restarting Ocius Sentinel after a software download this setting defines the delay in milliseconds between instructing Sentinel to shut down and restarting it.  | 3000   |
| offline.reconnect.retr<br>y.limit | When Ocius Sentinel reports that it is offline from the remote server this setting can be used to configure a number of connection retries. A value of -1 indicates unlimited retries. If a connection still cannot be established after the required number of retries then the <code>auto.offline</code> setting applies.   | 0  |
| gift.card.type                    | Defines the type of gift card supported by the core where<br>0 = Park Retail (the default)<br>1 = SVS<br><b>Note:</b> The POS may override this setting to specify the gift card type in its request message.   |  |
| report.card.events                | If true then the core will send <code>DeviceEvent</code> messages to the POS when a card is inserted into or removed from the PED. This is determined from status messages sent to the core by Ocius Sentinel.  | false  |

| Setting         | Description   | Default         |
|-----------------|---|-----------------|
| print.dcc.quote | If true then the core will print a DCC currency conversion quote at the point when the customer is asked to make a DCC decision at the PED.   | true            |
| keystore.name   | The name of the keystore file containing the key for decrypting passwords.<br>Since the keystore file will be created in the cores/OciusSentinel folder, the property can either include the relative path, or the keystore file can be copied to the base EFTLink folder.<br>Example with path<br>keystore.name = cores/OciusSentinel/ocius.keystore<br>Example where the keystore file has been copied to the base EFTLink folder<br>keystore.name = myfile.dat | ocius.keystore. |

## Translating and Suppressing Status Messages

Status messages sent by Ocius Sentinel for display at the POS can be translated or suppressed by adding entries to `ocius.properties`. Each message is identified by a number and the Ocius Sentinel integration guide lists all the possible messages.

As an example, status message 1 displays the text `Enter Gratuity`. To change this to “Enter Tip” the following entry can be added to `ocius.properties`:

```
status.1=Enter Tip
```

To suppress this message leave the text blank (nothing after the equal sign) as follows:

```
status.1=
```

## Overriding Other Text Messages

There are a number of other messages and prompts which are provided by the core itself and these are also configurable. The settings in `ocius.properties` are listed below with their defaults:

- `confirm.auth.code.prompt=Confirm Transaction?`
- `confirm.auth.code.yes.option=Yes - Confirm Txn`
- `confirm.auth.code.no.option=No - Decline Txn`
- `voice.referral.prompt=Call Auth Centre`
- `voice.referral.tel.text=Tel:`
- `voice.referral.mid.text=MID:`
- `voice.referral.tid.text=TID:`
- `voice.referral.amount.text=Amount: £`
- `voice.referral.trailing.text=`
- `voice.referral.yes.option=Authorise`
- `voice.referral.no.option=Abort`
- `voice.referral.auth.entry.prompt=Enter Auth Code (or blank to cancel)`
- `signature.verification.prompt=Valid Signature?`
- `signature.verification.yes.option=Yes - Confirm Txn`
- `signature.verification.no.option=No - Decline Txn`
- `signature.verification.reprint.option=`

- `cashback.prompt=Please enter cashback amount`
- `declined.card.removal.prefix.text= Declined -`
- `svs.partial.payment.title=PARTIAL PAYMENT ONLY`
- `svs.requested.amount.text=Requested £`
- `svs.available.amount.text=Available £`
- `svs.outstanding.amount.text=Outstanding £`
- `svs.partial.payment.yes.option=Continue`
- `svs.partial.payment.no.option=Cancel`

## Positioning Dialogue Options

For POS systems which support this it is possible to specify the position or order of some dialogue options using index numbers. The index should be an integer with value 1 or higher. The maximum index number allowed and the interpretation of the number will depend upon the implementation at the POS, for example in the case of Retail-J there are 8 button positions available down the right-hand side of the screen so the index numbers would range from 1 to 8.

The following settings are available:

```
confirm.auth.code.yes.position
confirm.auth.code.no.position
voice.referral.yes.position
voice.referral.no.position
signature.verification.yes.position
signature.verification.no.position
signature.verification.reprint.position
svs.partial.payment.yes.position
svs.partial.payment.no.position
```

## XML Receipts

Ocius Sentinel is able to supply raw receipt data in XML form rather than as formatted text. There are a considerable number of data fields available in this way (see the latest Ocius Sentinel Integration Guide for a full list). Here is an example of an XML signature receipt received by the core from Sentinel:

```

<VoucherDetails>
  <TrainingMode>>false</TrainingMode>
  <ReceiptType>Signature</ReceiptType>
  <Header>B & Q</Header>
  <PTID>PW001654</PTID>
  <TID>04380001</TID>
  <MID>21249872</MID>
  <MkTransactionID>1552313</MkTransactionID>
  <TxnDateTime>2010-12-06 20:40:37.845 CET</TxnDateTime>
  <CardScheme>Visa</CardScheme>
  <PAN>*****2222</PAN>
  <ExpiryDate>12/12</ExpiryDate>
  <TxnType>Sale</TxnType>
  <CaptureMethod>SWIPED</CaptureMethod>
  <CustomerPresent>>true</CustomerPresent>
  <ECommerce>>false</ECommerce>
  <ContAuth>>false</ContAuth>
  <AccountOnFile>>false</AccountOnFile>
  <PinEntered>>false</PinEntered>
  <CreditDebitMessage>Please debit my account</CreditDebitMessage>
  <CurrencySymbol>£</CurrencySymbol>
  <CurrencyAbbreviation>GBP</CurrencyAbbreviation>
  <Amount>1.00</Amount>
  <Total>1.00</Total>
  <CVM>Please Sign Below</CVM>
  <KeepText1>Please Keep This Receipt</KeepText1>
  <KeepText2>For your Records</KeepText2>
  <EFTSN>0508</EFTSN>
  <AuthCode>789DE</AuthCode>
  <Reference>101206 61 1 1</Reference>
  <Footer>B & Q</Footer>
  <GratuityBoxRequired>>false</GratuityBoxRequired>
  <ExtendedReceipt>>false</ExtendedReceipt>
  <DisableCurrencySymbol>>false</DisableCurrencySymbol>
  <AuthOnly>>false</AuthOnly>
  <CardSchemePrintText></CardSchemePrintText>
  <PrintAttempts>1</PrintAttempts>
  <ContactlessMSD>>false</ContactlessMSD>
  <TokenRegistrationResult>NotSet</TokenRegistrationResult>
  <TokenRegistrationOnly>>false</TokenRegistrationOnly>
</VoucherDetails>

```

In XML mode the core must be configured to convert the XML data into formatted text receipts. Formatting is achieved using template files in which free text and XML fields can be positioned and left, right or centre justified as required. Any number of templates can be created and you would typically expect to have seven or more, one for each of the merchant, signature, customer, merchant void, customer void, merchant declined and customer declined receipts, and further templates for any extended functionality (for example gift cards). Below is an example of a template file:



**customer\_template.txt**

```

<WIDTH=36>
<CENTRE>Customer Test Template
-----
Card Sale<RIGHT><Total>
<PAN>
-----
Card   : <CardScheme>
Number : <PAN><RIGHT><CaptureMethod>
AID    : <AID>
App Date : <AppEff>
Cryptogram : <CID>/<AC>
Auth Code : <AuthCode>
Merchant ID: <MID>
Terminal ID: <TID>
-----
<CreditDebitMessage>
<CENTRE><CVM>

```

In the template, XML element names are specified in angled brackets like this <CVM> and each will be substituted with the actual value supplied by Sentinel. There are four special directives used for formatting which are:

- <WIDTH=nn> This specifies the maximum width of the receipt in columns.
- <CENTRE> This will centre any text which appears after it on the same line.
- <RIGHT> This will right-justify any text which appears after it on the same line.
- <SUPPRESS> The receipt will not be printed.

---

**Note:** All of the above directives must be uppercase to be recognized.

---

In order to decide which template to use for a receipt the core will read a file called `ocius_receipt.properties` in which templates can be selected by looking for one or more values in the XML data. This file contains entries in the form

```
template-filename=<XML-element-1>required-value<XML-element-2>required-value
```

If all of the XML elements listed on the line have the specified value then that template file will be used. Below is an example file:

**ocius\_receipt.properties**

```
customer_template.txt=<ReceiptType>Customer
merchant_template.txt=<ReceiptType>Merchant
signature_template.txt=<ReceiptType>Signature
```

When looking for a match templates are checked in the order that they appear in `ocius_receipt.properties`. If no matching template is found then the core will return the entire XML data in place of a formatted receipt. If a template appears which does not specify any XML fields to match on (nothing after the equal sign) then that template will always be treated as a match.

It is also possible to match partial values using one or more of the flags [PREFIX], [SUFFIX] or [CONTAINS] followed by the partial text to match. For example:

```
contactless_template.txt= <ReceiptType>Customer<CaptureMethod>[SUFFIX] CONTACTLESS
```

The above will match when `ReceiptType` has the fixed value `Customer` and `CaptureMethod` is any text followed by `CONTACTLESS`.

---

## Keystore

The encryption key must be generated and stored in a keystore. To achieve this, the following steps must be followed:

### Windows Operating Systems

- Open a command prompt, and change directory to the eftlink location.
- Type: `encrypt.bat -k <keystore name> <properties file>`  
For example, `encrypt.bat -k ocius.keystore ocius.properties`.

Keystore file will be generated and stored in the data directory.

## Password Encryption

Default configuration requires `user.id`, `user.pin` and, where used, `account.id` and `transax.account.id` to be encrypted in `ocius.properties`.

`user.id`, `user.pin`, `account.id` and `transax.account.id` is allocated or configured in the Ocius Sentinel software itself, and varies from site to site.

To achieve this, the following steps must be followed:

### Windows Operating Systems

To encrypt a password; open a command prompt and change directory to eftlink's location.

- Type `encrypt.bat -e <keystore name> <properties file> <password>`.

For example, `encrypt.bat -e ocius.keystore ocius.properties` [followed by the required password as a final parameter].

- Password and initialization vector will be outputted to the console.

Copy and paste it to the appropriate property in `ocius.properties`.

To re-encrypt a password (or multiple passwords) with new encryption settings; open a command prompt and change directory to eftlink's location.

- Type `encrypt.bat -r <keystore name> <properties> <encrypted passwords colon separated> <previous initialization vectors colon separated> <keygen type> <cipher type> <key size> <iterations>`.

For example, `encrypt.bat -r ocius.keystore ocius.properties [Encrypted password1: Encrypted password2] [Encrypted password iv1: Encrypted password iv2] AES AES/CBC/PKCS5Padding 128 10000`.

- Re-encryption uses existing crypto settings in the properties file to decrypt the password. Once the password is decrypted, a new keystore file is generated using the new crypto parameters specified at the command line and the new encrypted password / initialization vector is generated.
- When using AES algorithm with a keysize that is greater than 128, you may get `java.security.InvalidKeyException: Illegal key size or default parameters`. If so, Additional Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files will need to be downloaded and extracted to `%JAVA_HOME%/jre/lib/security/`

## Supported Functions

The following operations are supported by this implementation of the Ocius Sentinel interface.

- Logon and logoff (at the beginning and end of a shift or trading period)
- Sale
- Refund
- Card Read (for non-EFT cards only)
- X reports (reconciliation)
- Z reports (reconciliation with closure)
- Customer receipt re-print (via maintenance menu)
- SVS gift cards

---

---

# VeriFone Point Scandinavia

## General Information

This document covers EFTLink Integration with VeriFone Point Scandinavia Payment Systems. It should be read in conjunction with the *Oracle Retail EFTLink Framework Installation and Configuration Guide*.

---

---

**Note:** VeriFone Point Scandinavia was formerly known as SteriaPay.

---

---

## Disambiguation

This VeriFone Point implementation is for use with the VeriFone Point Scandinavia Payment System, formerly known as SteriaPAY. There is also a Point US implementation, which is unrelated.

## EFTLink General

This document assumes static EFTLink configuration. When deploying with a POS that supports dynamic configuration, all property settings referred to below should be set on the POS, and not directly into local property files.

## System Architecture

EFTLink connects directly to the payment system using a proprietary socket protocol.

## Fileset

Verifone Point Scandinavia uses:

### Oracle files

cores/SteriaPay/steriapaycore.jar  
steriapay.properties

### VeriFone Point files

Obtain PayPoint.jar from Point Scandinavia, and place in the same folder (cores/SteriaPay) as steriapaycore.jar  
PayPoint.jar

---

---

**Note:** There is a UK interface within EFTLink to PayPoint, an online payment service provider, which uses a completely different paypoint.jar. Although that interface, and the associated jar file are not included in this release, the similarity in name may cause confusion, which is outside of Oracle's control.

---

---

## Language

There are no language specific translation files within steriapay.jar

## Core Classname

The following should have been set in EftLinkconfig.properties by installcore.bat or installcore.sh

```
EPSCore0 = manito.eft.steriapay.SteriaPayCore
```

## Configuration Settings

Configuration settings should be defined in steriapay.properties.

This will be copied from cores/SteriaPay to the base eftlink folder by installcore.bat or installcore.sh

The available settings are listed below (none of these are compulsory so the file may be absent, and the default steriapay.properties only has com.port=127.0.0.1). Each setting should be placed on a separate line in the file in form name=value. The setting names are all lower case.

| Setting                         | Description   | Default          | Example  |
|---------------------------------|---|------------------|--|
| com.port                        | The com port of the terminal if serial communications are being used (in the form COMn where n is the number of the port) or the IP address of the terminal for TCP/IP communications (in the form n.n.n.n).  | COM1             |  |
| baud.rate                       | The baud rate to use to communicate to the terminal for serial communications.  | 115200           |  |
| verify.signature                | When a signature is required, this specifies whether it should be verified as authentic.  | false            |  |
| signature.verification.question | When signature verification is required, this is the text of the question.<br><b>Note:</b> The terminal will deliver the signature receipt after approving the transaction. Therefore if the operator answers no to the signature verification question the core must request a reversal. For this reason after the operator answers no, both approval and reversal receipts will be printed. | Signature Ok?    |  |
| print.two.chip.card.receipts    | The terminal delivers one set of receipt text for a chip card transaction. This setting configures the core to print two copies of a chip card receipt (one each for merchant and customer).  | false            |  |
| receipt.one.title               | The title of the first receipt to be printed.   | no title printed | receipt.one.title=*** MERCHANT RECEIPT ***\n<br><b>Note:</b> The character sequence \n denotes a line break. |

| Setting  | Description   | Default                   | Example  |
|--|---|---------------------------|--|
| receipt.two.title                                | The title of the second receipt to be printed.<br>Note: If print.two.chip.card.receipts=false then use this setting to specify a title not receipt.one.title.   | no title printed          | receipt.two.title=*** CUSTOMER RECEIPT ***\n                               |
| signature.receipt.title                          | The title of the signature receipt.   | same as receipt.one.title |  |
| receipt.header                                   | The receipt header. This will print at the top of the receipt above the title.  | no header printed         | receipt.header=Retailer's Name\nRetailer's Address\nTel: +44 1234 123456\n |
| receipt.footer                                   | The receipt footer. This will print at the bottom of the receipt.<br><b>Note:</b> At the time of writing the current version of the java PayPoint software delivers some receipts with a paper cut message printed at the bottom, which cannot be changed or removed via PayPoint's own configuration. Since the core will insert the receipt footer below this it will be necessary to use the receipt.text.to.remove setting below to remove the paper cut message. | no footer printed         | receipt.footer=\nThank you for choosing\n     Retailer's name.             |
| receipt.text.to.remove                           | Allows a section of the receipt text provided by the terminal to be removed. This can be used to remove the paper cut message if required. The text to be removed should be specified exactly as shown on the receipt. The character sequence \n can be included to remove preceding or trailing line breaks.   | nothing removed           | receipt.text.to.remove=\n....PAPER CUT...                                  |
| reference.title                                  | This setting defines a label to print at the beginning of the line containing the reference field.<br><b>Note:</b> A label called REF: is already used elsewhere in the SteriaPay receipt.  | POS                       |  |
| remove.blank.lines.from.top.of.steriapay.receipt | The terminal may deliver receipts containing blank lines at the top. If adding a header and/or title it may be useful to remove these, which can be achieved using this true/false setting.   | false                     |  |

## Other Information

### Supported Functions

The following additional operations are supported by this implementation of the Verifone Point Scandinavia interface:

- Logon and logoff
- Sale (with cashback)
- Refund
- Reversal
- Reconciliation with closure
- Print stored reports (via the EFTLink maintenance menu function)
- Print last receipt (via the EFTLink maintenance menu function)

---

---

# Verifone Point (US)

## EFTLink General

This document assumes static EFTLink configuration. When deploying with a POS that supports dynamic configuration, all property settings referred to below should be set on the POS, and not directly into local property files. It should be read in conjunction with the *Oracle Retail EFTLink Framework Installation and Configuration Guide*.

## Disambiguation

This Point implementation is for use with Mx915 terminals in the US, with communication based on a socket/XML protocol. There is also a Point implementation in Norway, which is unrelated.

## Minimum Version

The Point interface requires a minimum EFTLink version of v1.1.124.

## System Architecture

Verifone Point is deployed as an intelligent terminal. EFTLink connects directly to the terminal using a proprietary socket/XML protocol.

## Fileset

In addition to standard EFTLink files, PointUS uses:

- `cores/pointus/pointuscore.jar` - executable code for the PointUS EFTLink core.
- `pointus.properties` - configuration settings to specify which features are enabled and to define communication parameters for the interface with the EFT terminal.

**Note** - If the POS supports dynamic configuration, properties can be set there instead of in `pointus.properties`

## Core Classname

`manito.eft.pointus.PointUSCore`

This should be set as `EPSCore<x>` in `eftlinkconfig.properties`.

## Configuration Settings

The full set of configuration properties is defined and commented in `pointus.properties`.

## Key Settings

Settings that may be different for each POS/PED.



| Setting    | Description           | Default | Example      |
|------------|-----------------------|---------|--------------|
| TerminalIP | IP of Mx915 terminal. |         | TerminalIP = |

## Secondary Settings

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

| Setting             | Description   | Default | Example                     |
|---------------------|---|---------|-----------------------------|
| Terminal address    | Port number   | 5015    | TerminalPort = 5015         |
| ResponseTimeout     | Time allowed in seconds for the transaction to complete at the terminal. This needs to be long enough to cover all customer interaction and host authorization.   | 120     | ResponseTimeout = 120       |
| ValidateLoyaltyData | When a loyalty card swipe is requested, the customer may identify themselves by entering a phone number rather than swiping a card. If loyalty cards are suitably defined in the card range file and tagged as "Loyalty", this can be checked.<br><br>Option to enable validation of loyalty data to try to differentiate between card numbers and phone numbers. | false   | ValidateLoyaltyData = false |

## Administration Functions

The terminal has some administration/maintenance functions. These are normally invoked from a dedicated "EFT Maintenance" button, but if this is not available, they could be accessed by an engineer using the EFTLink built-in test harness.

EFTLink uses DeviceProxy messages to display input prompts on the POS to manage these functions.

| Function             | Description  |
|----------------------|--|
| Terminal-POS Pairing | The terminal has to be paired with a specific POS, by entering a code.   |
| Registration         | This operation displays a 4-digit number on the POS that must then be typed into the terminal to complete the pairing. |
| Unregistration       | This operation removes a pairing.  |
| Test MAC             | This operation tests that the terminal is accessible and that a pairing is in place.                                   |
| Day Report           | Print a non-closing day report (summary)   |
| Day End              | Print a day report and close the current day. Manual alternative to automated ReconciliationWithClosure.               |

|                  |  |
|------------------|--|
| Last Transaction | Print details of the last transaction at the terminal. |
|------------------|--|

## Supported Functions

Below is a list of supported functionalities of the interface to Merchant Link. Many functionalities are provided by PointUS, such as Loyalty, Cashback etc. (please refer to interface specification for details) but are not implemented because of the business requirement.

| Function                    | Description   |
|-----------------------------|---|
| Payment                     | Sends payment request to the terminal. Terminal will return a response message with unformatted receipt strings for customer and/or merchant receipts.<br><br>In an event of referral where authorization cannot be obtained online then a prompt for authorization code will appear; authorization code must be obtained via telephone and entered here.<br><br>If successful, appropriate receipts will be printed at the end of transaction. |
| Reversal                    | Sends reversal request to the terminal. This will reverse a transaction specified by the transaction number, found on the receipt, which must be captured by the POS and pass on to EFTLink.  |
| Refund                      | Sends refund request to the terminal. This will refund a transaction with specified amount.   |
| Reconciliation / Settlement | This is supported directly by the terminal via TCP/IP request.  |
| Sale State Notifications    | Sends line items through to the device so the customer display can be updated in line with the POS.   |
| SVC Payment                 | Sends a Gift or Merchandise credit card payment request to the terminal. If there are not enough funds available, only the funds available will be deducted. The POS client will have to settle the transaction with another tender in this scenario.   |
| SVC Activate                | Sends a Gift or Merchandise credit card activation request to the terminal.   |
| SVC Deactivate              | Sends a Gift or Merchandise credit card deactivation request to the terminal. The account is disabled after this as the request is intended to be used for lost or stolen cards. It is not possible to use the card or account once this request has been issued and accepted.  |
| SVC Add Value               | Sends a Gift or Merchandise credit card add value request to the terminal. This will only add value to an account that has been activated.  |
| SVC Balance Enquiry         | Sends a Gift or Merchandise credit card balance enquiry request to the terminal.  |
| SVC Unload (Cashout)        | Sends a Gift or Merchandise credit card cash out request to the terminal. All funds are deducted from the account and the cash back amount is returned to the POS. The account is not deactivated as part of this process.  |



## General Information

### Overview

This document covers EFTLink Integration with WorldPay Payment Systems. It should be read in conjunction with the *Oracle Retail EFTLink Framework Installation and Configuration Guide*.

### System Architecture

EFTLink connects to the WorldPay application that is installed on the same PC as the POS, using a proprietary socket protocol. The WorldPay application must be started.

---

---

**Note:** This document does not cover the install of the WorldPay software.

---

---

### Fileset

The following files are used in the EFTLink folder:

`cores/WorldPay/worldpaycore.jar`

`worldpay.properties` (optional, if not present defaults apply)

### Language

There are no translation files in `worldpaycore.jar`. EFTLink Framework should be set to default English. See the *Oracle Retail EFTLink Framework Installation and Configuration Guide*, EFTLink General Information, Translation section:

`EftlinkConfig.properties`

`DisplayLanguage = EN`

### Core Classname

The following should have been set in the `EftlinkConfig.properties` file by `installcore.bar` or `installcore.sh`

`EPSCore0=manito.eft.worldpay.WorldPayCore`

### Configuration Settings

The core is configured via settings inserted into the `worldpay.properties` file located in the chosen EFTLink folder. If the default port numbers are used within WorldPay's software configuration then this file does not need to be present as the core will work without it. The available settings are listed below.

---

---

**Note:** The software was previously called YesPay.

---

---

| Setting                   | Description   | Default  | Example                           |
|---------------------------|---|----------|-----------------------------------|
| yeseft.folder             | The path to the folder where the WorldPay software is installed. Worldpay is normally installed in a folder at the root of the C: drive of the PC called YESEFT.  | \\YESEFT | yeseft.folder = \\YESEFT          |
| request.port              | The socket port for making payment requests   | 10000    | request.port = 10000              |
| receipt.port              | The socket port for receiving receipts.   | 20000    | receipt.port = 20000              |
| message.port              | The socket port for receiving status messages and dialogue requests.  | 8000     | message.port = 8000               |
| perform.card.range.lookup | If true, EFTLink will use its mapping file CardRange.xml to determine the card scheme name based on information returned by WorldPay. Otherwise it will return the text provided by WorldPay.   | false    | perform.card.range.lookup = false |
| embed.customer.receipt    | If true, EFTLink will return the customer receipt to the POS to be included in its own receipt rather than printing it separately.<br>Note: not all POS systems may support this feature.   | false    | embed.customer.receipt = false    |
| suppress.merchant.receipt | If true, EFTLink will discard the merchant receipt.   | false    | suppress.merchant.receipt = false |
| store.merchant.receipt    | If true, EFTLink will return the merchant receipt to the POS to be added to the electronic journal rather than printing it separately. This setting is overridden by suppress.merchant.receipt.<br>Note: not all POS systems may support this feature.  | false    | store.merchant.receipt = false    |
| language                  | The language code for translating responses from WorldPay on the message port.<br>The translations are taken from WorldPay files in the WorldPay folder. The default value is "en_GB", and references part of the filename provided by WorldPay.<br>JVIMessageBundle_en_GB.properties in C:\YESEFT\properties folder. | en_GB    | language = en_GB                  |

| Setting                  | Description  | Default                                     | Example   |
|--------------------------|--|---|---|
| signature.reprint.prompt | The text to display when asking if a signature receipt should be reprinted. This text will only be shown if the operator answers no when asked to confirm signature ok for a previous print.   | Blank, meaning reprint will not be offered. | signature.reprint.prompt =                            |
| notify.signature.print   | If true the POS will be notified that a signature receipt has been printed. This is for the business case where the signed receipt must be stored in the cash drawer and therefore the POS needs to know to open the drawer.<br><b>Note:</b> An additional setting is required in <code>EftlinkConfig.properties</code> to enable this function:<br><code>DeviceEvents=true</code> | true  | notify.signature.print = true                         |
| mid.text                 | The title to display for the merchant ID in voice referrals.   | MID:  | mid.txt = MID:  |
| tel.text                 | The title to display for the telephone numbers in voice referrals.   | Tel:  | tel.txt = Tel:  |
| auth.prompt              | The text to display for the authorization code entry prompts in voice referrals.   | Enter Auth Code (or blank to cancel)        | auth.prompt = Enter Auth Code (or blank to cancel)    |
| max.auth.code.length     | The maximum length allowed for an entered authorization code.  | 9   | max.auth.code.length = 9                              |
| cashback.prompt          | The text to display for the cashback prompt.   | Cashback required?                          | cashback.prompt = Cashback required?                  |
| cashback.amount.prompt   | The text to display for the cashback amount prompt   | Please enter cashback amount.               | cashback.amount.prompt = Please enter cashback amount |
| min.cashback             | This is the minimum cashback amount allowed.   | Blank (no minimum amount).                  | min.cashback =  |
| max.cashback             | This is the maximum cashback amount allowed.   | Blank (no maximum amount).                  | max.cashback = 100                                    |
| max.cashback.length      | This is the maximum length allowed for an entered cashback amount.   | 5   | max.cashback.length = 5                               |
| currency.symbol          | The currency symbol to use when displaying cashback limits to the operator. This can be any text required, for example, "GBP" and so on.   | £   | currency.symbol = £                                   |

| Setting          | Description  | Default               | Example                              |
|------------------|--|-----------------------|--------------------------------------|
| cnp.prompt       | This is the text to display for the customer not present prompt.   | CNP confirmation      | cnp.prompt = CNP confirmation        |
| response.timeout | The timeout in milliseconds to wait for a response from WorldPay after sending a request. It is recommended that this be left disabled (indefinite) and leave the timeout to WorldPay. | 0 (indefinite).       | response.timeout = 0                 |
| print.x.report   | Whether to print an X report on reconciliation.  | false                 | print.x.report = false               |
| print.z.report   | Whether to print a Z report on reconciliation with closure.  | false                 | print.z.report = false               |
| x.report.title   | The title for X reports.   | ** EFT X<br>REPORT ** | x.report.title=** EFT X<br>REPORT ** |
| z.report.title   | The title for Z reports.   | ** EFT Z<br>REPORT ** | z.report.title=** EFT Z<br>REPORT ** |

## Other Information

### Additional Functions

The following additional operations are supported by this implementation of the WorldPay interface:

- Sale
- Refund
- Refund with token
- Reversal
- X reports (reconciliation)
- Z reports (reconciliation with closure)

### Integration Notes

#### WorldPay configuration

The WorldPay software must be configured to use its socket interface on all three ports (request, receipt and message) respectively. Within the WorldPay (YESEFT) configuration utility the relevant tabs are Interfacing, Receipt and HostEvt.

#### Online/Offline indication

In a card payment response the miscellaneous data field will indicate whether the authorisation was online, offline or manual (voice referral). The format will be {Status=xxx} where xxx is one of ONLINE, OFFLINE or MANUAL.

---

## Device ID

The terminal number will be returned in the DeviceID element of the EFTLink login response (if the WorldPay software is running at the point of login) and with each card payment response thereafter. An example login response is provided below.

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse RequestType="Login" ApplicationSender="POSSIM" WorkstationID="1"
RequestID="2" OverallResult="Success">
  <Terminal DeviceID="12345678" />
</ServiceResponse>
```

---

**Note:** The Terminal Device ID should be the pertinent one for the terminal being connected.

---

## Refund with token

To perform refunds via token both the token and the card payment reference from the original sale must be provided in the refund request, please see below for an example of a payment response from EFTLink showing these fields.

```
<?xml version="1.0" encoding="UTF-8"?>
<CardServiceResponse RequestType="CardPaymentLoyaltyAward"
ApplicationSender="POSSIM" WorkstationID="1" RequestID="4"
OverallResult="Success">
  <Terminal TerminalID="22980092" DeviceID="0081226814" MerchantID="6818780"
STAN="345" />
  <Tender>
    <TotalAmount Currency="GBP">56.00</TotalAmount>
    <Authorization AcquirerID="UNKNOWN" TimeStamp="2015-04-29T12:45:31"
ApprovalCode="947265" CardType="3" Tender="0108" CardPAN="476173*****0119"
ExpiryDate="1251" CardCircuit="VISA CREDIT" TransactionReference="PGTR740971038"
/>
  </Tender>
  <CardValue CardType="3" Tender="0108" LoyaltyEligible="true">
    <CardPAN>476173*****0119</CardPAN>
    <EndDate>1251</EndDate>
    <CardCircuit>VISA CREDIT</CardCircuit>
    <Hash>1CCF57529637C314FBE9C6544BF10E3D16FE20B8</Hash>
    <Token>533173099D9A95649</Token>
    <TransactionReference>PGTR740971038</TransactionReference>
  </CardValue>
  <MiscellaneousData>{Status=ONLINE}</MiscellaneousData>
</CardServiceResponse>
```

Below is an example of a subsequent refund request from the POS.

```
<?xml version="1.0" encoding=" UTF-8"?>
<CardServiceRequest RequestType="PaymentRefund" ApplicationSender=" POSSIM "
WorkstationID="1" RequestID="5">
  <POSdata LanguageCode="en" SpooledPrint="false">
    <POSTimeStamp>2015-04-29T12:46:31</POSTimeStamp>
    <TransactionNumber>920</TransactionNumber>
  </POSdata>
  <TotalAmount Currency="GBP">56.00</TotalAmount>
  <CardValue>
    <Token>533173099D9A95649</Token>
    <TransactionReference> PGTR740971038</TransactionReference>
  </CardValue>
</CardServiceRequest>
```

---

**Note:** The Token and Transaction Reference in the above statement are demonstration values only.

---



## Reversal

Reversal requests require the card payment reference, PAN and card expiry date from the original transaction. Additionally a reversal should carry the same transaction number as the transaction it is cancelling. Below is an example reversal request with the necessary data fields highlighted.

```
<?xml version="1.0" encoding="UTF-8"?>
<CardServiceRequest RequestType="PaymentReversal" ApplicationSender="POSSIM"
WorkstationID="1" RequestID="9" RequestSubType="OperatorReversal">
  <POSdata LanguageCode="en">
    <POSTimeStamp>2015-06-09T11:48:29</POSTimeStamp>
    <TransactionNumber>401</TransactionNumber>
  </POSdata>
  <OriginalTransaction TerminalID="22980092" STAN="401" TimeStamp="2015-06-
09T11:48:27" RequestType="CardPaymentLoyaltyAward" ApprovalCode="956872"
MiscellaneousData="{Status=ONLINE}" />
  <TotalAmount Currency="GBP">15.00</TotalAmount>
  <CardValue CardType="3" Tender="0108" LoyaltyEligible="true">
    <CardPAN>476173*****0119</CardPAN>
    <EndDate>1263</EndDate>
    <CardCircuit>VISA CREDIT</CardCircuit>
    <Hash>52FDA2337F840BEE654353EA1D1F54FB5EFC2E98</Hash>
    <Token>533173099D9A95649</Token>
    <TransactionReference>PGTR327632569</TransactionReference>
  </CardValue>
</CardServiceRequest>
```

---

**Note:** The Transaction Number, Card Pan, End Date and Transaction Reference in the above statement are demonstration values only.

---

## Signature Print Notification

If the core is configured to notify the POS of a signature print (see section 0) then a device event will be generated as shown below. The POS should examine the EventType field to determine that this is a signature print notification.

```
<?xml version="1.0" encoding="UTF-8"?>
<DeviceRequest ApplicationSender="MICROS" WorkstationID="1" RequestID="5.11"
RequestType="Event">
  <Event EventType="SIGNATURE" />
</DeviceRequest>
```

The POS should acknowledge the device event as in the following example.

```
<?xml version="1.0" encoding="UTF-8"?>
<DeviceResponse RequestType="Event" ApplicationSender="MICROS" WorkstationID="1"
RequestID="5.11" OverallResult="Success" />
```