

Oracle® Retail EFTLink
Core Configuration Guide
Release 15.0
E69279-01

December 2015

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Primary Authors: Nigel Parish, Sutep Mahehyong, Tony Manda

Contributors: Phil Wells

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

Send Us Your Comments	ix
Preface	xi
Audience	xi
Documentation Accessibility.....	xi
Related Documents.....	xi
Customer Support.....	xi
Review Patch Documentation.....	xii
Improved Process for Oracle Retail Documentation Corrections	xii
Oracle Retail Documentation on the Oracle Technology Network.....	xii
Conventions.....	xii
Introduction	1
AJB FiPay	3
EFTLink General	3
Minimum Version.....	3
System Architecture.....	3
Fileset.....	3
Third party.....	3
Language.....	3
Core Classname.....	4
Configuration Settings	4
Key settings.....	4
Secondary Settings.....	4
Supported Functions	5
Banksys Core	7
General Information	7
EFTLink General	7
Minimum Version.....	7
System Architecture.....	7
Fileset.....	7
Language.....	7
Core Classname.....	7
Configuration Settings	8
Key settings.....	8
Secondary settings	8
Supported Functions	8
CCVPOS (CCV ITS)	10
EFTLink General	10
Minimum Version.....	10
System Architecture.....	10

Fileset	10
Language	10
Core Classname	10
Configuration Settings	11
Key settings	11
Secondary settings	11
Supported EFT Operations	12
Administration Menu	12
Ingenico	13
EFTLink General	13
System Architecture	13
Fileset	13
Language	13
EFTLink configuration	13
Configuration Settings	14
Functions supported	15
SixPay	16
General Information	16
EFTLink General	16
System Architecture	16
Fileset	16
Language	16
Core Classname	17
Configuration Settings	17
Key Configuration Settings	17
Optional Configuration Settings	17
Fixed Configuration Settings	18
Other Information	18
PED Identification/Selection	18
SolveConnect Core	19
General Information	19
EFTLink General	19
System Architecture	19
Fileset	19
Language	19
Core Classname	19
Configuration Settings	20
Key settings	20
Secondary configuration settings	20
Fixed configuration settings	21
Supported Functions	21
TransaxEFT	23

EFTLink General	23
Minimum Version.....	23
System Architecture.....	23
Fileset	23
Translation	23
Core classname.....	23
Configuration Settings	24
Key settings.....	24
Secondary settings	24
Fixed Settings	26
Supported Functions	26
VeriFone Ocius Sentinel.....	28
General Information	28
Overview	28
System Architecture	28
Fileset.....	28
Language.....	28
Core Classname.....	28
Configuration Settings	28
Mandatory Configuration Settings.....	29
Optional Configuration Settings.....	29
Translating and suppressing status messages	37
Overriding other text messages	37
Positioning dialogue options.....	38
XML receipts.....	38
Encrypting password files	41
Other Information	42
Functions supported.....	42
VeriFone Point Scandinavia.....	43
General Information	43
Disambiguation.....	43
EFTLink General	43
System Architecture.....	43
Fileset	43
Language.....	43
Core Classname.....	44
Configuration Settings	44
Other Information	46
Functions supported.....	46
Verifone Point (US)	47
EFTLink General	47
Disambiguation.....	47

Minimum Version.....	47
System Architecture.....	47
Fileset.....	47
Language.....	47
Core classname.....	48
Configuration Settings	48
Key settings.....	48
Secondary settings	48
Supported EFT Operations	49
Administration Menu.....	50
PointUS – Other Information	51
WorldPay.....	53
General Information	53
Overview	53
System Architecture	53
Fileset.....	53
Language.....	53
Core Classname.....	53
Configuration Settings	53
Other Information.....	57
Functions supported.....	57
Integration notes	57

Send Us Your Comments

Oracle Retail EFTLink, Core Configuration Guide, Release 15.0

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

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

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

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

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

Send your comments to us using the electronic mail address: retail-doc_us@oracle.com
Please give your name, address, electronic mail address, and telephone number (optional).

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

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

Preface

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 15.0 documentation set:

- *Oracle Retail EFTLink Release Notes*
- *Oracle Retail EFTLink Framework Installation Guide*
- *Oracle Retail EFTLink Security Guide*

Customer Support

To contact Oracle Customer Support, access My Oracle Support at the following URL: <https://support.oracle.com>

When contacting Customer Support, please provide the following:

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

Review Patch Documentation

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

Improved Process for Oracle Retail Documentation Corrections

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

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

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

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

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

Oracle Retail Documentation on the Oracle Technology Network

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

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

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

Conventions

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

This is a code sample

It is used to display examples of code

Introduction

After installing EFTLink from the *Oracle Retail EFTLink Framework Installation 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.

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 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 FiPay interface requires a minimum EFTLink version of v15.0

System Architecture

EFTLink connects to FiPay software; either running on a store server or locally on the POS PC.

Note: This document does not cover installation of AJB software

Fileset

In addition to standard EFTLink files, FiPay uses:

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

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 *EFTLink Framework Installation 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

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

`EPSCore0 = manito.eft.ajb.FiPayCore`

Configuration Settings

Settings are defined in `fiipay.properties`, which should have been copied from `cores/FiPay` to the base `eftlink` folder by `installcore.bat` or `installcore.sh`

Key Settings

These settings must be set for all POSs

host

IP address where FiPay is running. Default is localhost

Example

`host = 10.0.0.99`

store.number

The unique store number allocated by AJB

Example

`store.number = 100`

emvkeys.provided

If enabled, EMV initialization keys 'EMV_KEYS.DAT' must be provided and place in FIPAYEPS folder. This property is ignored if `emv.initialization` is false. The default is false.

Example

`emvkeys.provided = false`

enable.emv.initialization

If enabled, EMV initialization command will be sent at POS logon. An option in Administration options will also be available to allow adhoc initialization. The default is false.

Example

`enable.emv.initialization = false`

Secondary Settings

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

port

TCP/IP port number, default 24900

Example

```
port = 24900
```

creditdebit.prompt

Credit/Debit prompt, controls whether to prompt operator for the card type (debit or credit). Default is false, meaning FiPay will determine card type automatically.

Example

```
creditdebit.prompt = false
```

response.timeout

FiPay response timeout, specify the number of seconds to wait for response from FiPay, default 120

Example

```
response.timeout = 180
```

pos.validate.swipe

If enabled, after the card is swiped or inserted, the operator gets a message with the card scheme name for example Visa or MasterCard, asking if they wish to proceed. This is to allow the rejection of card types; for example Diners Club. The default is false.

Example

```
pos.validate.swipe = false
```

electronic.signature

If enabled, signature prompt will appear on the terminal and operator will get a prompt to confirm whether signature is ok. The default is true.

Example

```
electronic.signature = true
```

enable.signature.logging

If enabled, signature data will be logged in the log file. The default is false.

Example

```
enable.signature.logging = false
```

Note: This should be enabled for debugging purposes only.
As soon as the debugging is complete, set back to false.

enable.tokenization

To enable support for tokenization. The default is false.

Example

```
enable.tokenization = false
```

Supported Functions

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

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.

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 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

VIC_PORTNAME

Serial port name

Example

```
VIC_PORTNAME = COM1
```

LANGUAGE

Language code, default en (English): options fr = French, nl = Dutch

Example

```
LANGUAGE = en
```

COUNTRY

Country code, default UK.

Example

```
COUNTRY = UK
```

Secondary settings

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

VIC_AUTOLOGON

Automatically perform a log-on at initialization, default false.

Example

```
VIC_AUTOLOGON = false
```

VIC_MANDATORYVOUCHER

If receipt data is not available, receipt voucher will be created using the data from the response, default false.

Example

```
VIC_MANDATORYVOUCHER = false
```

VIC_PRINTWIDTH

Receipt print width, default 38 (characters)

Example

```
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.

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 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 *EFTLink Framework Installation 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

These settings must be set for all POSs

CCVServerIP

IP address of the CCV terminal.

Example

```
CCVServerIP = 10.0.0.99
```

CCVWorkstationId

Name by which the local POS is identified by the terminal and/or host. Default POS01.

Example

```
CCVWorkstationId = POS01
```

Secondary settings

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

CCVChannel1

TCP/IP port used by EFTLink to allow the CCV terminal to connect back to the POS. Default 4102.

Example

```
CCVChannel1 = 4102
```

CCVResponseTimeout

Time allowed in seconds for the transaction to complete at the terminal. Default 330. 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.

Example

```
CCVResponseTimeout = 330
```

EmbedCustomerText

Option to embed the customer EFT voucher within the POS receipt to save paper. Default false.

Note: this goes against the CCV general rule of guaranteeing customer printout, so this feature should not be enabled without prior approval by CCV.

Example

```
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. Default false.

Example

EmbedJournalText = false

ManagementMenuX

Specifies which administration function to show at position 'n' of the menu and sets a label for it. See section on Administration Menu.

Example

ManagementMenu1 = Reprint Last Transaction, REPRINT_LAST_TICKET

ManagementMenu2 = Reconciliation - shift totals, SHIFT_TOTALS

ManagementMenu3 = Reconciliation - close shift, CLOSE_SHIFT

ManagementMenu4 = List card types, GET_CARD_CIRCUITS

ManagementMenu5 = Version, VERSION

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 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 *EFTLink Framework Installation 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.

c3path

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.

Example

```
c3path = C:/Program Files/Ingenico/C3Generic/bin/c3inet.exe
```

ip.address

The IP address of C3, default 127.0.0.1

Example

```
ip.address = 10.0.0.5
```

ip.port

The IP port of C3. The default is 9518

Example

```
ip.port = 9518
```

confirmation_list

A list of strings separated by a comma [,] which determines whether a user acknowledgement is required.

Example

```
confirmation_list = APPEL PHONIE,<>,VALIDEZ,Montant,PAN,FIN VAL,APPEL TELECOL
```

Note: This list may require modification before it can be used in a Spanish configured Ingenico system.

comms_timeout

The timeout in seconds between EFTLink and C3

Example

```
comms_timeout = 120
```

totalisation_type

The report type when doing reconciliation – not currently used

Example

```
totalisation_type = C
```

default_currency

The currency being used, defined by using the ISO 4217 numerical code, for example Euro 978, US Dollar 840, Sterling 826

Example

```
default_currency = 978
```

Functions supported

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)

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 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 *EFTLink Framework Installation 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

SixpayServerIP

IP address of the store server running MPD, default 127.0.0.1

Example

```
SixpayServerIP = 10.0.0.50
```

SixpayWorkstationID

Optional Setting for specific WorkstationID, and to set the WorkstationID format. Note 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

Example

```
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, default false.

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.

Example

```
WorkstationIDPosBased = true
```

This would mean that for POS2 with the `SixpayWorkstationID = POS1` set above, messages to MPD would be from POS2.

Careful use of WorkstationID settings and overrides in both the POS and EFTLink should make it possible to deploy a standard `sixpay.properties` file across all POSs.

Optional Configuration Settings

These settings are normally left on defaults

SixpayChannel0

TCP/IP port used for primary channel to MPD, default 20002

Example

```
SixpayChannel0 = 20002
```

SixpayChannel1

TCP/IP port for device requests from MPD

Example

```
SixpayChannel1 = 20007
```

SixpayResponseTimeout

Timeout in seconds for EFTLink to wait for the response from MDP, default 300.

Example

```
SixpayResponseTimeout = 300
```

IncludeSaleItems

If enabled, sale item details are included in the payment request. Default is false.

Example

```
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, default false.

Example

```
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, default false.

Example

```
ElectronicJournal = false
```

SignatureCheckTag

Trigger tag/text to detect that a signature has been asked for and should be checked, default "Signature:"

Example

```
SignatureCheckTag=sign
```

SignatureCheckTimeout

Timeout for Signature OK? question, default 30 seconds

Example

```
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 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 *EFTLink Framework Installation 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`

SourceID

The POS specific identifier, allocated by retailer, to be unique across the retailer's estate.

Example

```
SourceID = DPOS0001
```

Store.ID

A 4-digit store identifier which forms part of the reference number assigned to each transaction. Default 9999.

Example

```
Store.ID = 1234
```

POS.ID

2 digit POS identifier which forms part of the reference number assigned to each transaction. Default 99

Example

```
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`

ServiceHost

- Hostname or IP address of SolveConnect service.
- Example
- `ServiceHost = 127.0.0.1`

TransactionTimeoutPeriod

Number of seconds to allow a transaction to complete, default 180.

Example

```
TransactionTimeoutPeriod = 180
```

CancellationTimeoutperiod

Maximum number of seconds the core will wait for a transaction response following a cancellation, default 30.

Example

```
CancellationTimeoutPeriod = 30
```

MaintenanceMenuTimeout

The number of seconds to wait for an option to be selected before dismissing the Maintenance menu, default 30.

Example

```
MaintenanceMenuTimeout = 45
```

AuditLoggingEnabled

Enable/Disable logging of transaction results to an audit log, default false.

Example

```
AuditLoggingEnabled = false
```

TransactionReferenceScheme

The format and source of Store and Till-ID values. Recognised values are `Properties` and `PowerPOS`. Default is `Properties`.

If set to `PowerPOS`, the `POS.ID` value will be automatically extracted from the POS system name at run time, so the setting in `SolveConnect.POS.properties` can be left at zero.

Example

```
TransactionReferenceScheme = PowerPOS
```

TransactionNumberFromPOS

Whether to use the transaction number from the POS (with suffixes to ensure uniqueness) rather than the default auto-incrementing number, default true.

Example

```
TransactionNumberFromPOS = true
```

ForcePurchaseWithCashback

Force all POS Purchase requests to be converted to Solve Purchase with Cashback requests, default true.

Example

```
ForcePurchaseWithCashback = true
```

PromptForCashbackCharge

Prompt for a cashback charge, default true.

Example

```
PromptForCashbackCharge = true
```

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.

Payment

Sends payment request to the terminal. Terminal 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; authorization code must be obtained via telephone and entered here. 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 passed on to EFTLink.

Refund

Sends refund request to the terminal. This will refund a transaction with a specified amount.

GiftCard

Sends giftcard payment request to the terminal. Specified amount will be deducted from the giftcard.

Administration options to add balance and check balance is also supported.

Receipt Reprint

Reprint merchant/customer receipt.

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 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

TransaxEFTChannel0

The TCP port on which the Core sends requests and device responses to TransaxEFT. Default is 8900.

Example

```
TransaxEFTChannel0 = 8900
```

TransaxEFTChannel1

The TCP port on which the Core listens for responses and device requests from TransaxEFT. Default is 9900.

Example

```
TransaxEFTChannel1 = 9900
```

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

ReceiptFormatFile

Name and path of the receipt XSLT translation file used to format the receipt text. Default is `transaxeft\AccreditationReceipt.xml`

Example

```
ReceiptFormatFile = transaxeft\AccreditationReceipt.xml
```

BalanceEnquiryFormatFile

Name and path of the card balance enquiry XSLT translation file. Default is `transaxeft\Balance.xml`

Example

```
BalanceEnquiryFormatFile = transaxeft\Balance.xml
```

ReconciliationFormatFile

Name and path of the reconciliation report XSLT translation file. Default is `transaxeft\Reconciliation.xml`

Example

```
ReconciliationFormatFile = transaxeft\Reconciliation.xml
```

ReceiptTextPassThroughEnabled

Only enable when TransaxEFT provides pre-formatted plain text receipt lines rather than name/value pairs. Default is false.

Example

```
ReceiptTextPassThroughEnabled = false
```

TransaxEFTResponseTimeout

Extend the period we will wait for a CardServiceResponse message after sending our request. Default is 120.

Example

```
TransaxEFTResponseTimeout = 120
```

TransaxEFTOperatorRecoverySupported

Do not allow the operator to specify the success or failure of a transaction. Default is false.

Example

```
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. Default is 60.

Example

```
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. Default is true.

Example

```
MaintMenuOptReceiptReprintEnabled = false
```

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. Default is true.

Example

```
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. Default is false.

Example

```
MaintMenuOptPEDTestEnabled = false
```

MaintMenuOptPEDTestPrintedEnabled

Enable/disable the menu option TXT_PED_TEST_PRINTER_OUTPUT. Set to true for Power POS. Default is true.

Example

```
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. Default is false.

Example

```
MaintMenuOptEODQueryEnabled = false
```

MaintMenuOptEODQueryPrintedEnabled

Enable/disable the menu option TXT_QUERY_LAST_RECONCILIATION_PRINTER_OUTPUT. Default is true.

Example

```
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.

Payment

EFTLink sends payment requests to Transax. Transax 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.

Refund

EFTLink sends refund requests to Transax. This will refund a transaction with specified amount.

Reconciliation / Settlement

Prints a settlement report at day end.

Reprint receipt

Reprints last 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 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 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 [Encrypting password files](#)

user.id

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.

Example

```
user.id=89eb96f2dfed02384e99fb7f8bfea610
```

user.pin

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.

Example

```
user.pin=89eb96f2dfed02384e99fb7f8bfea610
```

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.

ip.address

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.

ip.port

The IP port of the terminal. The default is 25000.

terminal.menu.configuration

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.

account.id

The account ID to send with each transaction. This option is used in some deployments, and Verifone would indicate the value to use. The default is blank.

auto.logon

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). The default is true.

pause.before.auto.logon

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. The default value for this setting is 1000.

auto.logon.pause

The number of milliseconds to wait after an auto logon before sending a transaction. The pause should be for several seconds.

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.

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.

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. The default is 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. The default is false.

max.cashback.length

The maximum length permitted for a cashback amount. The default is 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. The default is

```
*** Duplicate Receipt ***\n
```

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. The default is false.

offer.reprint

Whether to display the “Re-print/Continue” dialogue after printing a receipt. The default is true.

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. The default is 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:

- 0 – Not Set
- 1 – Do Not Register (the default)
- 2 – Register
- 3 – Register Only
- 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:

- 0 - Non EFT card
- 1 – EFT card
- 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 Encrypting password files
. The default is true.

auto.confirm.licence.key

If true (the default) then there will be an automatic response to the LicenceDetailConfirmation status from Ocius Sentinel.

card.wait.mode

If true the core will send CARDWAIT records, otherwise it will operate in standard mode. The default is 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. The default is 0 (no timeout).

wait.record.capture.methods

This is a hex bitmap of the capture methods that the PED is to allow.

The hex bitmap is comprised of the following hex values:

- Keyed = 01
- Swipe = 02
- ICC = 04
- Reserved = 08

The default is for the core to leave this blank, in which case Sentinel will apply the following default:

- ICC + Swipe + Keyed = 07

wait.record.fallback.methods

This is a hex bitmap of the fallback methods that the PED is to allow.

The hex bitmap is comprised of the following hex values:

- Fallback from ICC to Swipe = 01
- Fallback from Swipe to Key = 02

The default is for the core to leave this blank, in which case Sentinel will apply the following default:

- Fallback from ICC to Swipe + Fallback from Swipe to Key = 03

auto.offline

If true the core will automatically instruct Ocius Sentinel to work offline if the remote server is unavailable. The default is false.

reference

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:

- Date: the transaction date provided by the POS in the form YYMMDD
- Time: the transaction time provided by the POS in the form HHMMSS
- Transnum: the transaction number provided by the POS
- User: the operator ID provided by the POS when it logged on to EFTLink

-
- `pos`: the POS ID provided by the POS when it logged on to EFTLink

The default value of this setting is:

```
date transnum user pos
```

CARDWAIT with CNP

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:

```
wait.record.capture.methods.cnp = 1
```

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.

simple.cnp.enabled

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. The default is false.

Note: In this mode `<CNP>true</CNP>` is added to the XML receipt data for telesales.

transax.account.id

The account ID to use for Transax transactions.

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. The default value is 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:

```
transax.declined.operator.message=Transax Payment Void
```

or it may be used to display one of the fields of a Transax XML receipt. For example:.

```
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. The default is false.

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.

Note: `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. The default is 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. The default is 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. The default is 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. The default is false.

space.out.status.messages

Indicates whether status text from Ocius Sentinel should be spaced out for display, for example `ExpiryDateRequired` becomes `Expiry Date Required`. The default is 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. The default is true.

cancel.card.wait.delay

When `card.wait.mode=true` 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.

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. The default is 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 `await.ready.after.transaction=false`.

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). The default is 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. The default is 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. The default is 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. The default is 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. The default is 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. This is B&Q specific functionality. The default is 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. The default value is `MERCHANT COPY`.

fixed.receipt.customer.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. The default value is `CARDHOLDER COPY`.

fixed.receipt.signature.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. The default value is `Please Sign Below`.

fixed.receipt.void.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 void receipt. The default value is `VOID`.

fixed.receipt.declined.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. The default value is `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. The default value is -1 which indicates unlimited retries.

cancel.download.on.failure

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. The default value is true. 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.

ocius.sentinel.exe.path

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. The default value is as follows:

```
\Program Files\Verifone\Ocius Sentinel\OciusSentinel.exe
```

ocius.sentinel.restart.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. The default value is 3000.

offline.reconnect.retry.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. The default value is 0. If a connection still cannot be established after the required number of retries then the `auto.offline` setting applies.

gift.card.type

Defines the type of gift card supported by the core where

- 0 = Park Retail (the default)
- 1 = SVS

Note: 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 `DeviceEvent` 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. The default is false.

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. The default is true.

keystore.name

The name of the keystore file containing the key for decrypting passwords.

The default file name is `ocius.keystore`.

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.

Example with path

```
keystore.name = cores/OciusSentinel/ocius.keystore
```

Example where the keystore file has been copied to the base EFTLink folder

```
keystore.name = myfile.dat
```

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.

Encrypting password files

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` will be allocated or configured in the Ocius Sentinel software itself, and will vary from site to site.

A utility is available for encrypting a list of user IDs and passwords contained in a file. The utility will read an input file containing unencrypted data and will generate an output file with the corresponding encrypted data using a keystore file that should be distributed to allow EFTLink Ocius Sentinel Core to decrypt the data to communicate to the Ocius Sentinel software.

This file (`encryptor.jar`) is in the same directory as `ociussentinelcore.jar`

The utility is accessed via a command prompt in the same directory, creates or uses keystore files, input and output files in the same directory.

This software has 2 functions:

- To create a keystore file.
- To encrypt IDs, PINs etc. from an input file into an output file.

`Encryptor.jar` does not decrypt previously encrypted ID, PINs and so on.

Steps

Step1

Choose a 16 character encryption key and generate a keystore file. The keystore file must be available on the deployed system in order to decipher the encrypted data.

For example the following command will store a key (`TheEncryptionKey`) in a default file called `ocius.keystore`:

```
java -cp encryptor.jar Encryptor -k TheEncryptionKey
```

Note: `TheEncryptionKey` should be a 16 character string chosen by the installer / integrator. It does not need to be remembered

It is recommended that an alternative keystore file name is used. This is created by using the command as follows:

```
java -cp encryptor.jar Encryptor -k TheEncryptionKey myfile.dat
```

When an alternative keystore file name is used then the name must be included as an extra parameter at the end of the encryption command specified below, and must also be configured in `ocius.properties` for example
`keystore.name=cores/OciusSentinel/myfile.dat`

Step 2

Generate encrypted IDs, PINs and so on.

This will use an input file – for example `input.txt` – and generate an output file – for example `output.txt`.

The input file should contain one or more lines consisting of five fields; store ID, user ID, user PIN, account ID and Transax account ID. These must be separated by a single space. Below is an example of an input file:

input.txt

```
000998 1234 1234 12345678 1
```

```
000999 2345 2345 23456789 2
```

```
java -cp encryptor.jar Encryptor -e input.txt output.txt
```

If you are using an alternative keystore name add the name as an extra parameter

```
java -cp encryptor.jar Encryptor -e input.txt output.txt myfile.dat
```

This will generate the following file (the lines are shown wrapped, but actually each store will be on one line, and the store number is not encrypted):

output.txt

```
000998 370b391e25f4db85757e53f97defd91d 370b391e25f4db85757e53f97defd91d
fa39801a915c23cf4bb2d6d31955c25c e7a93f877734f4ea89deea1720f49a82
000999 3020f704f7f42200b7bada7f1a1a462f 3020f704f7f42200b7bada7f1a1a462f
3f45b42077d9f087a965e5b52c8394cf c1fa53da87ccc459151764fe49657d7e
```

Therefore the encrypted string 370b391e25f4db85757e53f97defd91d can be written to ocius.properties:

```
user.id=370b391e25f4db85757e53f97defd91d
```

```
user.pin=370b391e25f4db85757e53f97defd91d
```

This task may well be done by a company centrally referencing a number of store IDs , and the keystore file distributed with the encrypted IDs, PINs and so on for entry in each site's ocius.properties

Note: If you are doing this on site, delete input.txt afterwards, as otherwise the security is compromised.

Other Information

Functions supported

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 *EFTLink Framework Installation 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.

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).

Default = COM1

baud.rate

The baud rate to use to communicate to the terminal for serial communications.

Default = 115200

verify.signature

When a signature is required, this specifies whether it should be verified as authentic.

Default = false

signature.verification.question

When signature verification is required, this is the text of the question.

Default = Signature Ok?

Note: 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.

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).

Default = false

receipt.one.title

The title of the first receipt to be printed.

Default = no title printed

Example:

```
receipt.one.title=*** MERCHANT RECEIPT ***\n
```

Note: The character sequence `\n` denotes a line break.

receipt.two.title

The title of the second receipt to be printed.

Default = no title printed

Example:

```
receipt.two.title=*** CUSTOMER RECEIPT ***\n
```

Note: If `print.two.chip.card.receipts=false` then use this setting to specify a title not `receipt.one.title`.

signature.receipt.title

The title of the signature receipt.

Default = same as `receipt.one.title`

receipt.header

The receipt header. This will print at the top of the receipt above the title.

Default = no header printed

Example:

```
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.

Default = no footer printed

Example:

```
receipt.footer=\nThank you for choosing\n      Retailer's name.
```

Note: 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.

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.

Default = nothing removed

Example:

```
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.

Default = POS:

Note: A label called REF: is already used elsewhere in the SteriaPay receipt.

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.

Default = false

Other Information

Additional Functions supported

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 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 Scandinavia, which is unrelated.

Minimum Version

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

System Architecture

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

Note: The POS has to be registered/paired with the terminal before any transactions can be processed – See the section on Administration Functions.

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.

Language

There is a single English translation file in `pointus.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 and edited.

```
LangEN_PointUS.properties
```

The language file used will follow the language set in the EFTLink framework; see the *EFTLink Framework Installation Guide*, EFTLink General Information, Translation section, so EFTLink Framework must be set to English (the default).

```
EftlinkConfig.properties
```

```
DisplayLanguage = EN
```

Core classname

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

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

Configuration Settings

Settings are defined in `pointus.properties`.

Key settings

These settings must be set for all POSs.

TerminalIP

IP address of the Point terminal.

Example

```
TerminalIP = 10.0.0.99
```

EmbeddedPrinting

Option to embed the customer EFT voucher within the POS receipt to save paper. Default `false`.

Example

```
EmbeddedPrinting = false
```

Secondary settings

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

ConnectionTimeout

Time allowed in seconds to make a connection to the payment terminal. Default 10.

Example

```
ConnectionTimeout = 10
```

ResponseTimeout

Time allowed in seconds for the transaction to complete at the terminal. Default 120. This needs to be long enough to cover all customer interaction and host authorization.

Example

```
ResponseTimeout = 120
```

ConnectionTimeout

Time allowed in seconds to make a connection to the payment terminal. Default 10.

Example

```
ConnectionTimeout = 10
```

ShowlineItemsOnVoucher

Option to include details of each constituent item on the printed EFT voucher. Default `false`. Normally this would be handled by the POS.

Example

```
ShowlineItemsOnVoucher = false
```

ShowEMVTagsOnVoucher

Option to include diagnostic data relating to ICC payments on the printed EFT voucher. Default false. This option may need to be enabled for bank approvals/accreditation.

Example

```
ShowlineItemsOnVoucher = false
```

MaxLineItemTextLength

Maximum displayable item description length. Default 17. Long item description will be truncated to this length.

Example

```
MaxLineItemTextLength = 17
```

SignatureCheckFloorLimit

Value limit above which a signature will be required for non-PIN transactions. Default \$0.00.

Example

```
SignatureCheckFloorLimit = 0.00
```

SignatureCheckTimeout

Time allowed in seconds for the operator to verify the customer's signature. Default 30.

Example

```
SignatureCheckTimeout = 30
```

ValidateLoyaltyData

Option to enable validation of loyalty data to try to differentiate between card numbers and phone numbers. Default false.

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.

Example

```
ValidateLoyaltyData = false
```

AdminMenu0.n

Specifies which administration operation to show at position 'n' of the administration menu and sets a label for it. See the section on Administration Menu.

Example

```
AdminMenu0.1 = TXT_REGISTER, Register  
AdminMenu0.2 = TXT_UNREGISTER, Unregister  
AdminMenu0.3 = TXT_TEST_MAC, Test MAC  
AdminMenu0.4 = TXT_LAST_TRAN, Last Transaction  
AdminMenu0.5 = TXT_DAY_REPORT, Day Report  
AdminMenu0.6 = TXT_DAY_END, Day End  
AdminMenu0.7 = TXT_SIGNATURE_CAPTURE, Signature Capture  
AdminMenu0.8 = TXT_CANCEL, Cancel
```

Supported EFT Operations

The following operations are supported by this implementation of the VeriFone Point US interface.

Payment/Refund

Payment and refund by credit/debit card.

Signature Capture

If the payment/refund response indicates that a signature is needed, EFTLink will initiate a signature capture and present it to the operator for verification.

Reversal/Void

Payment/refund transactions can be cancelled either on explicit request from the POS or automatically on signature verification failure.

Linelist Display

The terminal can be used to show a running summary of the items and/or discounts included in the sale.

Reconciliation / Settlement

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

Administration Menu

The Point 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.

In particular, the POS has to be registered/paired with the terminal, by entering a code number on the terminal.

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.

Day End

Print a day report and close the current day.

Last Transaction

Print details of the last transaction at the terminal. This can be used to recover payment details in the event of a system failure.

PointUS – Other Information

There are a number of configurable settings on the Mx915 terminal that need to be set to the correct values for this core to function properly. Refer to VeriFone documentation for details on how to access the terminal configuration menus.

- `sigcapture N` – disable integrated signature capture

Note: Disabling integrated signature capture is required to allow 3-Byte ASCII electronic signature capture to function correctly.

- `sigimagetype 3BA` – set the signature capture data type to 3-Byte ASCII
- `devtypesuffix P` – enable chip & PIN processing

Also, receipt header (`header_N`) and footer (`footer_N`) lines should all be removed. The POS adds its own standard header and footer to each print operation.

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 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 *EFTLink Framework Installation 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.

yeseft.folder

The path to the folder where the WorldPay software is installed.

Default value: `\YESEFT` (Worldpay is normally installed in a folder at the root of the C: drive of the PC called YESEFT)

Example

```
yeseft.folder = \YESEFT
```

request.port

The socket port for making payment requests, default value: 10000

Example

```
request.port = 10000
```

receipt.port

The socket port for receiving receipts, default value: 20000

Example

```
receipt.port = 20000
```

message.port

The socket port for receiving status messages and dialogue requests, default value: 8000

Example

```
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. Boolean, the default value is false.

Example

```
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. Note: not all POS systems may support this feature. Boolean, the default value is false.

Example

```
embed.customer.receipt = false
```

suppress.merchant.receipt

If true, EFTLink will discard the merchant receipt. Boolean, the default value is false.

Example

```
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`. Note: not all POS systems may support this feature. Boolean, the default value is false.

Example

```
store.merchant.receipt = false
```

language

The language code for translating responses from WorldPay on the message port.

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.

JVTMessageBundle_en_GB.properties in C:\YESEFT\properties folder.

Example

```
language = en_GB
```

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. The default value is blank, which means reprint will not be offered.

Example

```
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. Boolean; the default value is true.

Example

```
notify.signature.print = true
```

Note: An additional setting is required in
EftlinkConfig.properties to enable this function:
DeviceEvents=true

mid.text

The title to display for the merchant ID in voice referrals, default value is MID:

Example

```
mid.txt = MID:
```

tel.text

The title to display for the telephone numbers in voice referrals. The default value is Tel:

Example

```
tel.txt = Tel:
```

auth.prompt

The text to display for the authorization code entry prompts in voice referrals. The default value is Enter Auth Code (or blank to cancel)

Example

```
auth.prompt = Enter Auth Code (or blank to cancel)
```

max.auth.code.length

The maximum length allowed for an entered authorization code. The default value is 9

Example

```
max.auth.code.length = 9
```

cashback.prompt

The text to display for the cashback prompt. The default value: Cashback required?

Example

```
cashback.prompt = Cashback required?
```

cashback.amount.prompt

The text to display for the cashback amount prompt. The Default value: Please enter cashback amount.

Example

```
cashback.amount.prompt = Please enter cashback amount
```

min.cashback

This is the minimum cashback amount allowed. The default value is blank (no minimum amount).

Example

```
min.cashback =
```

max.cashback

This is the maximum cashback amount allowed. The default value is blank (no maximum amount).

Example

```
max.cashback = 100
```

max.cashback.length

This is the maximum length allowed for an entered cashback amount. The default value is 5.

Example

```
max.cashback.length = 5
```

currency.symbol

The currency symbol to use when displaying cashback limits to the operator. This can be any text required, e.g. "GBP" and so on. The default value is £.

Example

```
currency.symbol = £
```

cnp.prompt

This is the text to display for the customer not present prompt. The default value is CNP confirmation.

Example

```
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. The default value is 0 (indefinite).

Example

```
response.timeout = 0
```

print.x.report

Whether to print a X report on reconciliation. Boolean, the default value is false.

Example

```
print.x.report = false
```

print.z.report

Whether to print a Z report on reconciliation with closure. Boolean, the default value is false.

Example

```
print.z.report = false
```

x.report.title

The title for X reports. The Default value is `** EFT X REPORT **`

Example

```
x.report.title=** EFT X REPORT **
```

z.report.title

The title for Z reports. The default value is `** EFT Z REPORT **`

Example

```
z.report.title=** EFT Z 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" />
```