

## Guardian

### Installation/Configuration/Integration Guidelines

Last Updated: Friday, 09 March 2018



## Table of Contents

<b>Guardian .....</b>	<b>1</b>
<b>Installation/Configuration/Integration Guidelines .....</b>	<b>1</b>
Introduction .....	4
Fully Integrated configuration .....	4
Semi Integrated configuration .....	5
Minimum System Requirements .....	6
Installation .....	6
Configuration .....	8
General Configuration.....	8
Admin Configuration.....	9
Server Configuration .....	10
Client Configuration .....	14
TMS Settings .....	16
Developers Guide.....	17
Classes available in the API .....	17
TransactionHook Class.....	18
TransactionInfo Class .....	19
TillInformation Class .....	21
TransactionHook Sample Code .....	22
ReceiptHook Class.....	23
TokenTransactionHook .....	24
TokenTransactionHook Sample Code .....	24
VoidTransactionHook.....	26
NonGuiTransactionHook.....	27
CustomerValidation .....	29
ReportHook.....	30
ReceiptProvider.....	31
ReceiptProvider Sample Code.....	31
TaxfreeHook – Tax Free Shopping .....	33
TaxfreeInfo .....	34
TaxfreeHook Code Sample.....	34
IFSF .....	36
Partial Fuel Authorisation .....	37

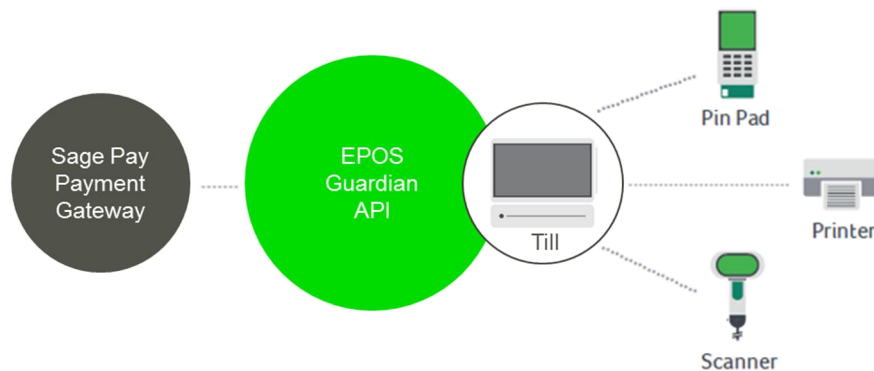
Testing Utilities .....	37
Guardian Tester .....	38



Legal Notice This installation and configuration document ("Manual") has been prepared to assist you with integrating your own (or your client's) service with Sage Pay's payment gateway via our Guardian software. You are not permitted to use this Manual for any other purpose. Whilst we have taken care in the preparation of this Manual, we make no representation or warranty (express or implied) and (to the fullest extent permitted by law) we accept no responsibility or liability as to the accuracy or completeness of the information contained within this Manual. Accordingly, we provide this Manual "as is" and so your use of the Manual is at your own risk. In the unlikely event that you identify any errors, omissions or other inaccuracies within this Manual we would really appreciate it if you could please send details to us using the contact details on our website at [www.sagepay.com](http://www.sagepay.com). We may update this Manual at any time without notice to you. Please ensure that you always use the latest version of the Manual, which we publish on our website at [www.sagepay.com](http://www.sagepay.com), when integrating with our payment gateway. Copyright © Sage Pay Europe Limited 2016. All rights reserved.

## Introduction

This guide contains essential information on how to install and implement Sage Pay's card processing software, **Guardian**. Guardian provides a simple interface for easy card payment integration to EPOS systems. It can be configured to operate in semi-integrated mode, fully-integrated mode and IFSF listening service mode.



## Fully Integrated configuration

In fully-integrated mode, the PED (pin entry device e.g. Verifone VX820) is serially or USB connected to the till system. Guardian interacts with the PED for card entry, pin entry etc. Guardian software on the till will establish a secure connection to Sage Pay's payment gateway. Guardian will also print card receipts to the standard receipt printer connected to the till that is also used by the EPOS software for sales receipt printing. In fully-integrated mode the EPOS software is integrated directly with the Guardian dll (dynamic link library).

### How it works:

- When a customer is ready to pay, the EPOS system will pass the value of transaction to a method (function) exposed by the Guardian DLL.
- The Guardian will pop up a window on top of the EPOS software which will provide information/data entry capability to the shop operator.
- Guardian will establish a secure TCP/IP connection to Sage Pay's payment gateway.
- Guardian will establish a secure connection to the serially/USB connected PED.
- Guardian will drive the transaction on the PED and onward into the payment gateway.
- Once the transaction has been complete, Guardian will print the merchant and customer copy receipts (as required) to the EPOS receipt printer.
- Guardian will return to the EPOS application indicating whether the transaction has been authorised or not. It will also provide (non-sensitive) transaction detail back to the EPOS software e.g. card scheme name / pin verified / signature verified / first six digits of card number/ last four digits of card number.

### Semi Integrated configuration

In semi-integrated mode, a fully functional PED is serially or TCP/IP connected to the till. The terminal has its own independent IP comms capability and as such, it establishes the secure TCP/IP connection to the payment gateway and not Guardian. The PED has an inbuilt receipt printer and normally will print the card receipts instead of Guardian (via the receipt printer attached to the EPOS till).

### How it works:

- When a customer is ready to pay, the EPOS system will pass the value of transaction to a method (function) exposed by the Guardian DLL.
- The Guardian will pop up a window on top of the EPOS software which will provide information/data entry capability to the shop operator.
- Guardian establishes a serial or TCP/IP connection to the PED.
- Guardian sends the transaction value to the PED which then comes to life and prompts for card insertion/card tapping etc.
- The PED establishes a secure TCP/IP connection to the payment gateway.
- Once the transaction is complete, the PED will print the merchant and customer receipts (as appropriate) if configured to do so.
- PED will then pass the result back to the Guardian software on the EPOS till.
- If Guardian is configured to print card receipts then it will do so via the receipt printer connected to the till.
- Guardian will return to the EPOS application indicating whether the transaction has been authorised or not. It will also provide (non-sensitive) transaction detail back to the EPOS software e.g. card scheme name / pin verified / signature verified / first six digits of card number/ last four digits of card number.

## Minimum System Requirements

Windows 7 +

Microsoft .Net framework 4.6.1 +

## Installation

To install Guardian

1. Copy the zip file to the local drive on the machine (C Drive)
2. Extract the files and the following folder structure will be created:

```
C:\sagepay\guardian\  
C:\sagepay\guardian\bin\  
C:\sagepay\guardian\config\  
C:\sagepay\guardian\docs\  
C:\sagepay\guardian\logfiles\  
C:\sagepay\guardian\receipts\  
C:\sagepay\guardian\reports\  
C:\sagepay\guardian\scripts\  
C:\sagepay\guardian\tms\  
C:\sagepay\guardian\ VcRedistributable\  
C:\sagepay\guardian\VerfioneDrivers\
```

3. The folder structure must not be changed as Guardian will expect that the above folder structure exists.
4. If you wish to install Guardian on a drive other than the default C drive, then there are a number of other changes which must be made.
  - a. Browse to the sagepay\guardian\bin\  
b. Right click on the Sagepay.WinApp.GuardianConfiguration.exe.config file and open through notepad  
c. In the <appSettings> section, change the value of the "defaultdrive" key to the required value  

```
<add key="defaultdrive" value="C:\"/>
```

  
d. Repeat step c in the Sagepay.WinApp.GuardianTester.exe.config
5. Along with Guardian dlls, the bin folder also contains two applications:
  - a. Integral.WinApp.GuardianConfiguration.exe (configure Guardian software)
  - b. Integral.WinApp.GuardianTester.exe (EPOS emulator application for testing)
6. If the EPOS application must run as a 32bit application then:
  - a. Install the Microsoft VC 2015 Runtime distributable (32bit)
  - b. Copy C:\sagepay\guardian\bin\cryptpp\_m\x86\ cryptpp\_m.dll to C:\sagepay\guardian\bin\ (overwriting the existing dll).
7. If the EPOS application must run as a 64bit (or targets 'Any CPU') application then:
  - a. Install the Microsoft VC 2015 Runtime distributable (64bit)
  - b. Copy C:\sagepay\guardian\bin\cryptpp\_m\x64\ \*.dll to C:\sagepay\guardian\bin\ (overwriting the existing dll(s)).
8. Microsoft VC 2015 Runtime distributables can be installed from :  
c:\sagepay\guardian\vc redistributable\

9. Please note that if enabling Guardian for OPOS printing then EPOS application **MUST** run as a 32-bit application.

## Configuration

Utility: c:\sagepay\guardian\bin\integral.winapp.guardianconfiguration.exe. This utility is used to fully configure the Guardian software.

### General Configuration

The first screen displayed is the General Configuration Tab, this tab allows general information about the merchant to be set. This information will be printed on receipts if Guardian is operating in IFSF mode. If Guardian is operating in semi or fully integrated mode then the merchant name and address will be passed programmatically by the EPOS software. See below for a full list of fields and explanations.

The screenshot shows the 'Guardian Configuration' window with the 'General Configuration' tab selected. The window has a menu bar with 'General Configuration', 'Admin Configuration', 'Server Configuration', 'Client Configuration', 'TMS Settings', and 'About'. Below the menu bar is a toolbar with icons for help, undo, redo, and save. The main area is divided into two sections: 'General' and 'Till Information'. The 'General' section contains a single field 'IFSFEnabled' with the value 'False'. The 'Till Information' section contains five fields: 'MerchantName' (Sage Pay Ireland), 'Address1' (50-51 Patrick Street), 'Address2' (Dun Laoghaire), 'Address3' (Co Dublin), and 'PhoneNumber' (01-2311777). At the bottom of the window is the 'sage Pay' logo and two buttons: 'Cancel' and 'Finish'.

Field	Value
IFSFEnabled	False
MerchantName	Sage Pay Ireland
Address1	50-51 Patrick Street
Address2	Dun Laoghaire
Address3	Co Dublin
PhoneNumber	01-2311777

Field	Description
IFSFEnabled	Specifies whether or not this is an IFSF Enabled Till
MerchantName	Name of Merchant
Address1	Merchant's Address Line 1
Address2	Merchant's Address Line 2
Address3	Merchant's Address Line 3
Address4	Merchant's Address Line 4

PhoneNumber

Merchant's Telephone Number

## Admin Configuration

The screenshot shows the 'Guardian Configuration' window with the 'Admin Configuration' tab selected. The 'Admin Settings' section is expanded, showing a table with the following values:

PedLanguage	en826
LogLevel	127
DefaultDrive	C:\

Below the table, the 'PedLanguage' field is highlighted, showing 'Ped Language' as a description. At the bottom of the window, the 'sage Pay' logo is displayed on the left, and 'Cancel' and 'Finish' buttons are on the right.

Field	Description
PedLanguage	In Fully integrated mode, the ped supports multiple languages. If the ped needs to be operated in a language other than English then Sage Pay will advise on the the value to be configured.
LogLevel	Indicates the level of logging in use by Guardian. This value should be left at the default setting unless a change is requested by Sage Pay support
DefaultDrive	Not in use at present

## Server Configuration

This configuration tab allows you to specify how Guardian will interact with the sagepay payment gateway.

The screenshot shows the 'Guardian Configuration' window with the 'Server Configuration' tab selected. The window has a title bar and several tabs: 'General Configuration', 'Admin Configuration', 'Server Configuration', 'Client Configuration', 'TMS Settings', and 'About'. Below the tabs is a toolbar with icons for help, undo, redo, and save. The main area is divided into two sections. The top section is titled 'General' and contains a list of configuration items with their values. The bottom section is titled 'ListenerPort' and contains a description of the port. At the bottom of the window is the 'sage Pay' logo and two buttons: 'Cancel' and 'Finish'.

General	
ListenerPort	1314
XSDLocation	C:\sagepay\guardian\bin\
IP Address of the Guardian Client machine	127.0.0.1
ClientPortNumber	1315
LogFileLocation	C:\sagepay\guardian\logfiles\
LogFileName	<b>GuardianServer.log</b>
MerchantsAcquirer	<b>Streamline</b>
AvailableCurrency	<b>GBP</b>
AvailableFuelType	<b>Topaz</b>
ConfigMode	<b>Fully Integrated</b>
SemiIntegratedTCPIPAddress	
SemiIntegratedTCPIPPort	
ReceiptConfirmMode	Transaction confirmation/receipt printing on till
ReceiptInfoMode	No receipt information lookup
InstallationType	<b>TEST</b>
SagepayLiveID	

**ListenerPort**  
Port on which Guardian Server will listen for incoming TCP/IP connections

**sage Pay**

Cancel Finish

Field	Description
Listener Port	When IFSF is enabled this specifies the port number that Guardian service will listen for incoming TCP/IP connections from the EPOS software.
XSDLocation	Folder where the IFSF XSD's (Schema Definition's) are located.
ClientIPAddress	In IFSF mode, this field defines the IP address of the till that the EPOS software is running on
ClientPortNumber	In IFSF mode, this field defines the TCP/IP port number that the EPOS software will be listening in order to receive device requests from Guardian service.
LogFileLocation	Location to which Guardian will write it's log file
LogFileName	Name of log file that Guardian will create
MerchantsAcquirer	Select the merchant's acquiring bank. This is the bank that the merchant has an agreement with for credit card processing.
MerchantsCurrency	If the merchant a GBP (UK based) or EUR (Irish based) merchant
AvailableFuelType	If the merchant is enabled for fuel card processing then then select the merchant's fuel profile, e.g. Topaz site / Texaco site etc.
ConfigMode	Select which mode (semi or fully) integrated that Guardian will operate as. Possible options are: "Fully Integrated" – serial ped "Semi Integrated (Serial)" – serial terminal "Semi Integrated (TCPIP)" – tcp/ip terminal "Semi Integrated (TCPIP) (Print on Till)" – tcp/ip terminal "Semi Integrated (TCPIP) (Rendered Receipts)" – serial terminal "Semi Integrated (Serial) (Print on Till)" – tcp/ip terminal "Semi Integrated (Serial) (Rendered Receipts)" – serial terminal
SemiIntegratedTCPIPAddress	If operating in a semi integrated mode, this is the IP address of the terminal
SemiIntegratedTCPIPPort	If operating in a semi integrated mode, this is the port number that the terminal is listening on
InstallationType	This field defines whether the installation is in test mode or live mode.
SagePayLiveID	This field is only configured when the Installation Type is in Live Mode. The ID for the site, which is supplied by Sage Pay, should selected here
SagePayTestID	This field is only configured when the Installation Type is in Test Mode. The ID for the site, which is supplied by Sage Pay, should selected here
EposType	Guardian can be used with both Microsoft RMS till applications and also with non RMS till applications. Select the till application type here.
PayrouterMerchantID	For each merchant, Sage Pay will provide a unique PayRouter Merchant Id which should be inserted here.
PayrouterClientID	For each till associated with a merchant, Sage pay will provide a unique client id. Value provided must be entered here.

DeviceType	If operating in fully integrated mode, select the ped that will be connected to the till.
ComPort	The serial communications port that the ped is connected to.
SplitCardTendering	Guardian has the ability to split payment across multiple credit cards. If 'Card split tendering is controlled by EPOS application' then the EPOS software will send the individual credit card amounts out to Guardian one at a time. If 'Card split tendering is controlled by Guardian, then the till sends the full amount of the transaction to Guardian and Guardian will allow the full amount to be split across a maximum of 3 credit cards.
TenderValuesAsDecimal	Specify whether Tender values are entered as decimal value or cent on the front end of Guardian
CardSwipeLocation	This configuration option is now redundant. All card swiping will be done on the ped or card card terminal
CustomerNotPresentEnabled	Are customer not present (mail order) transaction supported. Please note that key entry still required a ped or a card terminal to be connected to the
AutoAcceptonPinVerification	If this option is set to True, then if a transaction is pin verified then Guardian will automatically disappear and will not give the operator the option to cancel the transaction after it is authorised. If this this option is set to False, then the operator will be given an option to cancel a pin verified transaction after it is authorised.
AutoAcceptTappendNoVerification	Automatically close Guardian if transaction is tapped and there is no customer verification required, i.e do not give operator option to cancel (reverse) the transaction on the front-end of Guardian.
PrintCustomerReceiptFirst	If transaction is pin verified or no verification required, then this option dictates whether customer copy receipt is printed before merchant copy.
DailyTMSCall	If turned on, then after the first transaction of the day is performed, Guardian will connect to Sage Pay do check for any updates for the ped or offline configuration on the till.
ObscureCardNumber	Force Guardian to obscure the card number on merchant copy of receipts.
DccEnabled	Enable Guardian for dynamic currency conversion
CashbackType	If turned on, this specifies whether the cashback amount is entered in decimal format or in the lowest denominaton
E2eeEnabled	Enable a secure channel for delivery of sensitive card details between ped and Sagepay's hosted environment.
P2PEEnabled	If merchant is enabled for point to point encryption then this option must be set to TRUE, otherwise false.
PartialFuelAuthorisation	This parameter is used to control Guardian behaviour when it comes to fuel card processing. FALSE: If some or all of the product codes send through from the POS for a transaction are not allowed for the proffered fuel card then Guardian will fully reject the transaction and return error back to POS. It will provide a list of the fuel pridcyts that

	<p>are not allowed that the POS is trying to sell.</p> <p>TRUE: If some of the products sent through by the POS are not allowed for the proffered fuel card then Guardian will processed the allowed products and will return to POS the products that were not allowed and obviously the reduced value of the transaction that was processed.</p>
AutoRunReports	This configuration option is now redundant
OfflineModeEnabled	This setting dictates if Guardian can operate in offline mode (in the situation where the till cannot connect to Sage Pay's payment gateway).
CloseBatch	If this option is enabled then when Guardian does a z-report, it will close the batch of transactions with the payment gateway before running the actual z-report
ReprintReceipt	This configuration option is not currently supported
Receipt Printing Section	Self-Explanatory
TaxfreeProvider	Select the merchant's Taxfree Provider, if Taxfree is not provided then select - No Taxfree
TaxfreeMerchantID	For each merchant, Sage Pay will provide a unique Taxfree Merchant Id which should be inserted here.
TaxfreeClientID	For each till associated with a merchant, Sage pay will provide a unique Taxfree client id. Value provided must be entered here.
TaxfreeProductFileLocation	Select the location where the Taxfree product file location is located. This should be the same folder as the rest of the configuration items (sagepay\guardian\config\)
PrintTaxfreeForm	Self-Explanatory
TaxfreePrintMerchantReceipt	Self-Explanatory

## Client Configuration

This configuration tab allows you to customise how the Guardian front end will be displayed. It will also allow you to set the printer settings for Receipt Printing.

Guardian Configuration

General Configuration | Admin Configuration | Server Configuration | **Client Configuration** | TMS Settings | About

**Colours**

ClientBackgroundType: **Linear GradientBrush background**

ClientStartColour: **0, 128, 0**

ClientEndColour: **255, 255, 255**

ClientSolidColour: **255, 0, 0**

ClientBackgroundImage: C:\Users\Public\Pictures\Sample Pictures\Hydrangeas.jpg

**General**

ListenerPort: 1315

Folder in which IFSF XSD's are located: C:\sagepay\guardian\bin\

ServerIPAddress: 127.0.0.1

ServerPortNumber: 1314

LogFileLocation: C:\sagepay\guardian\logfiles\

LogFileName: **GuardianClient.log**

Type of Receipt Printer: **Print Direct Serial**

WindowsPrinter: HP LaserJet P3010 Series (10.0.0.206), Arial, False, 11

OPOSPrinterName: 12345

DirectPrinter: **COM6, 9600, NONE, 8, 1, None, 1B64040D0A1B690D0A**

**ListenerPort**

Port on which Guardian Client will listen for incoming TCP/IP connections

**sage Pay**

Cancel Finish

Field	Description
ClientBackgroundType	Select the Background Type of Guardian; one of Linear Gradient Brush (2 colours fading into each other), Solid Brush (Single Colour) or Image
ClientStartColour	When Linear Brush is selected this is the colour on the left of the background
ClientEndColour	When Linear Brush is selected this is the colour on the right of the background
ClientSolidColour	When Solid Brush is selected this is the colour of the background
ClientBackgroundImage	Defines the location of the background image when Image is in use, this is a browsable field.
ListenerPort	When IFSF is enabled this specifies the port number that Guardian Client will listen for incoming TCP/IP connections
XDSLocation	Folder where the IFSF XSD's (Schema Definition's) are located.
ServerIPAddress	IP Address of the Guardian Server machine
ServerPortNumber	TCP/IP Port on Server Machine which Guardian Client will connect to

LogFileLocation	Location of the Guardian log file
LogFileName	Name of the Guardian log file
Type of Receipt Printer	Type of Receipt Printer (Windows, OPOS or Direct Serial Printer)
WindowsPrinter	If receipts are to be printed to a Windows installed printer, then select the Printer from the dropdown and also specify the font to be used.
OPOSPrinterName	If printing to an OPOS installed printer, then specify the POS printer device name here.
DirectPrinter	If printing direct to serial (or LPT printer) then configure serial printer settings here
RMSReceiptPrinterType	If this is a Microsoft RMS till, then specify the RMS active printer number of the printer to use for receipt printing.
LineFeedsBeforePapercut	Some printers may cut the paper in such a way that part of the merchant copy of the credit card receipt is cropped. This dropdown can be used to alleviate this problem. The default value for this field is 3 which seems to cater for the vast majority of printers.
CutPaperAfterMerchantReceipt	Self-Explanatory
CutPaperAfterCustomerReceipt	Self-Explanatory
AppendLineFeed	Depending on the type of printer being used and the driver being used for the printer an extra linefeed character may have to be appended in order to get the printer to move to a new line.
ClientDisplayLogo	Specify whether or not to display a sagepay logo on the Guardian front end
Position	Display where on screen to display the Guardian front end screen
Size	Specify the size of the Guardian front end
TaxfreeSettings	Not in use currently

## TMS Settings

This tab allows you to configure daily TMS Calls on the PED

The screenshot shows the 'Guardian Configuration' window with the 'TMS Settings' tab selected. The window has a menu bar with 'General Configuration', 'Admin Configuration', 'Server Configuration', 'Client Configuration', 'TMS Settings', and 'About'. Below the menu bar is a toolbar with icons for help, undo, redo, and save. The main area is divided into two sections. The top section is titled 'General' and contains a table with the following data:

Field	Value
TMSCallType	Scheduled TMS call as Windows Task
TaskName	Daily TMS Call
TaskDescription	Run Daily TMS Call
TaskStartTime	11PM - 1AM

The bottom section is titled 'TMSCallType' and contains the text 'Select when a TMS call should be performed'. To the right of this text is a button labeled 'Run TMS Call Now'. At the bottom of the window is the 'sage Pay' logo and two buttons labeled 'Cancel' and 'Finish'.

Field	Description
TMSCallType	Select when a TMS call should be performed (As a scheduled task on windows, after first transaction, or allow IFSF to perform the call)
TaskName	Enter a Name for the Windows Task
TaskDescription	Enter a description of the Task
TaskStartTime	Select a time when the Windows Task should be scheduled

## Developers Guide

The following section describes the development interface provided to developers who develop their own till applications.

The API that the EPOS integrates with comes in the form of a Windows .NET dll. The dll is called Integral.Library.GuardianClient and is located in the sagepay\guardian\bin\ folder.

### Classes available in the API

Class	Description
TransactionHook	This class is used to process a card based transaction
TransactionInfo	If a transaction has been processed successfully (either authorised or declined) via the TransactionHook object, then this object will contain non-sensitive information relating to the transaction
TillInformation	This class is used to hold information related to the site and the till, and is used when Guardian prints credit card receipts
ReceiptHook	This class is used to print duplicate credit card receipts
TokenTransactionHook	This class is used to process a token based transaction
ReportHook	This class is used to run x and z reports
VoidTransactionHook	This class is used to void a credit card Transaction
NonGuiTransactionHook	This class is used to drive transactions from service devices
CustomerValidation	This class is used to perform account verification of customer
ReceiptProvider	This class can be used to dynamically get at receipts during the transaction. This would be used by EPOS software that want to print receipts itself.
TaxfreeHook	This class is used to process Taxfree transactions after a credit card transaction has completed successfully
TaxfreeInfo	If a Taxfree transaction has successfully completed, then this object will contain information relating to the Taxfree transaction
RunTaskHook	This class is used to perform specific tasks

## TransactionHook Class

```
Public Enum TRANSACTIONHOOK_TRANSACTIONTYPE
    INT_TT_SALE = 0
    INT_TT_REFUND
    INT_TT_PREAUTH
    INT_TT_FINALAUTH
    INT_TT_PREAUTHPARTIAL
    INT_TT_FINALAUTHADVISE
End Enum

Public Function Process(ByVal transactionType As TRANSACTIONHOOK_TRANSACTIONTYPE,
    ByVal transactionAmount As Integer,
    ByRef tillInfo As TillInformation,
    ByRef transInfo As TransactionInfo) As Boolean

Public Function Process(ByVal transactionType As TRANSACTIONHOOK_TRANSACTIONTYPE,
    ByVal transactionAmount As Integer,
    ByVal referenceText As String,
    ByRef tillInfo As TillInformation,
    ByRef transInfo As TransactionInfo) As Boolean

Public Sub InitialiseProductInfo()

Public Sub AddProductInfo(ByVal productCode As String,
    ByVal amount As String,
    ByVal litres As String,
    ByVal quantity As Integer)
```

	Notes
1	Transaction amounts must be passed in lowest denomination of currency
2	TillInformation object must be populated with site name and address (for receipt purposes) before calling the Process() method
3	The Process() method will return TRUE if the transaction was authorised otherwise it will return false.
4	If fuel cards are accepted on the then the EPOS software must call InitialiseProductInfo() followed by AddProductInfo() for each product being sold as part of the transaction. This must be done before calling the Process() method
5	When calling the AddProductInfo() method, the product code should be a Sage Pay defined, 3-digit product code
6	If the product being sold is a wet stock product then the litre quantity should be to decimal places but the decimal point should not be part of the value. E.g. 12.59 Litres would be passed as "1259"

## TransactionInfo Class

Parameter	Description
AuthorisationCode	If transaction was authorised then this property will contain the auth code
TransactionRefNo	Unique transaction reference number returned by Integral for the transaction
ResponseCode	This property can be used to determine the outcome of the transaction after the Process () method returns. 0 – transaction aborted by the user 1 – Transaction Authorised 2 – Transaction Declined 3 – Transaction Cancelled (at accept/reject stage)
PanFirst6Digits	First 6 digits of card number
PanLast4Digits	Last 4 digits of card number
SchemeName	Card scheme name. E.g. uk maestro solo laser visadebit electron maestro visa mastercard amex diners jcb debit mastercard
EmvAppicationLabel	EMV application label e.g. VISA DEBIT
EmvAppicationId	EMV application identifier
MerchantNo	Banking merchant number
TerminalId	Banking terminal id
CurrencyCode	Currency code of transaction, e.g. GBP
CustomerVerification	How was the customer verified: 0 – No verification 1 – Pin 2 – Signature 3 – Pin and Signature 4 - Card Holder Device
CashbackAmount	If cashback was accepted by the customer, then this field will contain the cashback value in lowest denomination of merchant's currency.
DataEntryMethod	How was the card processed: 0 – No data entry method 1 – Card was key entered 2 – Card was swiped 3 – Card was chipped (ICC) 4 – Tapped
CardClass	1 - Credit Debit 2 - Fuel Card

FuelBunkerCard	Is this card a fuel bunker card (True/False)
PrintFuelBunkerTotal	If the fuel card is a bunker card, then this property indicates if to print totals for customer copy of receipt for bunker products.
CustomerMileage	For a fuel transaction this property may contain the customer's odometer reading which should be printed on customer copy of receipt
CustomerRegNo	For a fuel transaction this property may contain the customer's registration number, which should be printed on customer copy of receipt.
CustomerReceipt	If transaction successful processed (i.e. no aborted by user) then this property will contain the customer receipt text with carriage return / linefeed characters inserted as appropriate.
Token	Token Number
PanLength	Length of card number used
CvvResult	If CVV checking enabled as part of transaction then this property will contain one of the following values: 0 – No CVV result 1 – CVV value matched 2 – CVV value not matched 3 – CVV value not checked
AvsResult	If AVS checking enabled as part of transaction then this property will contain one of the following values: 0 – No AVS result 1 – AVS not matched 2 – AVS not checked 3 – Partial AVS match
DccCurrencyCode	If DCC was accepted by the customer as part of the transaction then this field will contain the 3-character ISO alpha code that is the customer's home currency
ExchangeRate	If DCC was accepted by the customer then this field will contain the exchange rate between the customer's currency and the merchant's currency
DccAmount	IF DCC was accepted by the customer then this field will contain the value of the transaction in the customer's currency.
CardHolderName	Card holder's name
DisallowedFuelProducts	Colon separated list of products that are not sellable for the proffered fuel card.
TaxfreeEligible	Indicates whether or not this transaction is eligible for Taxfree (True/False)

### TillInformation Class

Parameter	Description
MerchantName	Merchant name for receipt header
Address1	Merchant Address 1 for receipt header
Address2	Merchant Address 2 for receipt header
Address3	Merchant Address 3 for receipt header
Address4	Merchant Address 4 for receipt header
PhoneNumber	Phone Number for receipt header
SiteNumber	Site number for receipt header
TillNumber	Till Number for receipt header
ReceiptNumber	Receipt number for receipt header
OperatorId	Operator ID for receipt header

## TransactionHook Sample Code

```
Imports Integral.Library.GuardianClient

Public Function ProcessTransaction() As Boolean

    Dim returnValue As Boolean = False
    Dim tillInformation As New TillInformation
    Dim transactionInfo As New TransactionInfo
    Dim transaction As New TransactionHook
    Dim transactionType As TransactionHook.TRANSACTIONHOOK_TRANSACTIONTYPE

    'Populate the till information object
    tillInformation.MerchantName = "Test Shop"
    tillInformation.Address1 = "Test Address 1"
    tillInformation.Address2 = "Test Address 2"
    tillInformation.Address3 = "Test Address 3"

    'If the till supports the processing of fuel cards then add
    'the products details associated with the transaction
    transaction.InitialiseProductInfo()
    transaction.AddProductInfo("002", "10900", "10000", 0) 'Unleaded
    transaction.AddProductInfo("045", "500", "", 1) 'Car Wash

    'Process the transaction
    transactionType = TransactionHook.TRANSACTIONHOOK_TRANSACTIONTYPE.INT_TT_SALE

    If transaction.Process(transactionType, 11400, tillInformation, transactionInfo) Then

        'Transaction has been authorised
        'TransactionInfo object will contain information about the transaction
        returnValue = True

    Else

        'Transaction has been aborted / cancelled / declined.
        'Check transactionInfo.ResponseCode to determine the exact outcome

    End If

    Return returnValue

End Function
```

### ReceiptHook Class

```
Public Function Process(ByRef tillInformation As TillInformation,  
                       ByVal transactionReference As Integer) As Boolean
```

Method return TRUE on success, otherwise false on failure

## TokenTransactionHook

```
Public Enum TRANSACTIONHOOK_TRANSACTIONTYPE
    INT_TT_SALE = 0
    INT_TT_REFUND
    INT_TT_PREAUTH
    INT_TT_FINALAUTH
    INT_TT_PREAUTHPARTIAL
    INT_TT_FINALAUTHADVISE
End Enum

Public Enum TOKENTRANSACTIONHOOK_TRANSACTIONMODE
    INT_TM_CUSTOMERPRESENT = 0
    INT_TM_CUSTOMERNOTPRESENT
    INT_TM_CONTINUOUSAUTHORITY
    INT_TM_ECOMMERCE
End Enum

Public Function Process(ByVal transactionType As TRANSACTIONHOOK_TRANSACTIONTYPE,
    ByVal transactionMode As TOKENTRANSACTIONHOOK_TRANSACTIONMODE,
    ByVal transactionAmount As Integer,
    ByVal token As String,
    ByVal expiryDate As String,
    ByVal referenceText As String,
    ByRef tillInformation As TillInformation,
    ByRef transactionInfo As TransactionInfo) As Boolean
```

	Notes
1	Value of transaction in lowest denomination. Please not that refund amount MUST be passed as positive values in the lowest denomination.
2	Token number. This will have been returned previously as part of a card transaction.
3	Token expiry date (MMYY format)
4	This TillInformation object will contain information specific to the site till which will be used for the header section of printed receipts
5	After the transaction has been processed, the TransactionInfo object will contain non-sensitive detail about the transaction.

## TokenTransactionHook Sample Code

```
Imports Integral.Library.GuardianClient
Imports Integral.Library.GuardianClient.TransactionHook
Imports Integral.Library.GuardianClient.TokenTransactionHook

Public Function ProcessTokenTransaction() As Boolean

    Dim returnValue As Boolean = False
```

```

Dim tillInformation As New TillInformation
Dim transactionInfo As New TransactionInfo
Dim transaction As New TokenTransactionHook
Dim transactionType As TRANSACTIONHOOK_TRANSACTIONTYPE
Dim transactionMode As TOKENTRANSACTIONHOOK_TRANSACTIONMODE
Dim token As String

'Populate the till information object object
tillInformation.MerchantName = "Test Shop"
tillInformation.Address1 = "Test Address 1"
tillInformation.Address2 = "Test Address 2"
tillInformation.Address3 = "Test Address 3"

'Process the transaction
transactionType = TRANSACTIONHOOK_TRANSACTIONTYPE.INT_TT_SALE
transactionMode = TOKENTRANSACTIONHOOK_TRANSACTIONMODE.INT_TM_CUSTOMERNOTPRESENT
token = "98634621845295476234"

If transaction.Process(transactionType, transactionMode, 11400, token, "1219", "test ref", tillInformation, transactionInfo) Then

    'Transaction has been authorised
    'TransactionInfo object will contain information about the transaction
    returnValue = True

Else

    'Transaction has been aborted / cancelled / declined.
    'Check transactionInfo.ResponseCode to determine the exact outcome

End If

Return returnValue

End Function

```

### VoidTransactionHook

Public Function Process(ByRef tillInformation As TillInformation,  
ByVal transactionReference As Integer) As Boolean

Method return TRUE on success, otherwise false on failure

	Notes
1	This TillInformation object will contain information specific to the site till which will be used for the header section of printed receipts
2	Transaction Reference Number of transaction to void. If value is –1 then Guardian will void last processed transaction. A unique transaction reference is returned to the EPOS software as part of process a transaction (via the TransactionInfo object).

## NonGuiTransactionHook

This class is used by self-service devices to drive transactions through Guardian in a non-GUI context.

```
Public Enum NONGUITRANSACTION_DATAENTRY
    DATAENTRY_CHIPORSWIPE = 0
    DATAENTRY_CHIPORSWIPEORTAP
End Enum

Public Enum NONGUITRANSACTION_RETURNCODE
    RETURNCODE_SUCCESS = 0
    RETURNCODE_FAILED
    RETURNCODE_CASHBACKALLOWED
End Enum

Public Enum NONGUITRANSACTION_CONFIRMTYPE
    CONFIRMTYPE_AUTHORISED = 0
    CONFIRMTYPE_DECLINED
    CONFIRMTYPE_CANCELLED
End Enum

Public Function StartTransaction(ByRef tillInformation As TillInformation) As Boolean

Public Sub EndTransaction()

Public Sub AbortTransaction()

Public Function AuthoriseTransaction(ByVal cashbackAmount As Integer,
                                     ByRef transactionInfo As TransactionInfo) As
NONGUITRANSACTION_RETURNCODE

Public Function CardEnquiry(ByVal transactionType As
TransactionHook.TRANSACTIONHOOK_TRANSACTIONTYPE,
    ByVal transactionAmount As Integer,
    ByVal allowedDataEntryTypes As NONGUITRANSACTION_DATAENTRY) As
NONGUITRANSACTION_RETURNCODE

Public Function ConfirmTransaction(ByVal confirmType As NONGUITRANSACTION_CONFIRMTYPE,
    ByVal transactionReference As String,
    ByVal authCode As String,
    ByRef transactionInfo As TransactionInfo) As Boolean

Public ReadOnly Property ErrorMessage() As String
Public ReadOnly Property MerchantReceipt() As String
Public ReadOnly Property StatusMessage() As String
Public ReadOnly Property TransactionReference() As String
```

	Notes
1	The client begins the transaction by calling the StartTransaction() method. The client must pass a populated TillInformation object to the method
2	<p>If StartTransaction() returns TRUE then the client should proceed to the CardEnquiry stage. The client should pass the type of transaction it wants to process, the value of the transaction and whether data entry should be chip/swipe only or chip/swipe/tap.</p> <p>If the CardEnquiry() method returns indicating RETURNCODE_CASHBACKALLOWED then the client can prompt for a cashback amount and add it to the transaction at the authorisation stage.</p>
3	<p>If cardEnquiry() returns RETURNCODE_SUCCESS or RETURNCODE_CASHBACKALLOWED then the client should call AuthTransaction(). The client should pass a cashback amount if required along with an empty TransactionInfo object.</p> <p>If AuthTransaction() returns RETURNCODE_SUCCESS then the TransactionInfo object passed to the method must be interrogated to determine if the transaction has been authorised or not.</p> <p>The MerchantReceipt() property will give the client the formatted merchant copy receipt which it can print (if required),.</p> <p>The TransactionReference() property will contain a unique transaction reference that will be used by the client to confirm the transaction.</p>
4	<p>The ConfirmTransaction() method must be used to confirm the transaction as 'accepted' (settle this transaction) or as 'cancelled' (reverse/void this transaction).</p> <p>The TransactionReference() property value must be passed to the ConfirmTransaction() method.</p>
5	The AbortTransaction() method can be called to abort the transaction. Once the transaction gets to the authorisation stage with the Guadian software, the transaction cannot be aborted but would have to be cancelled via the ConfirmTransaction() method after the AuthTransaction() method returns
6	The StatusMessage() property can be used to get real time status messages such as 'WAITING FOR CARD INSERTION' etc. A timer event on the client is the best way to call the StatusMessage() property.

## CustomerValidation

This class is used to perform customer account verification. A zero-value authorisation (with or without CVV number / post code) is sent online for authorisation (account verification) to the acquirer/issuer.

```
Public Function Validate(ByRef transactionInfo As TransactionInfo,  
                        ByRef tillInformation As TillInformation,  
                        ByVal postCode As String,  
                        ByVal promptForCvv As Boolean) As Boolean  
  
Public Sub AbortTransaction()  
  
Public ReadOnly Property StatusMessage() As String
```

	Notes
1	Client must pass in an empty TransactionInfo object. After Validate() method returns, this object will contain transaction detail that can be interrogated to determine if customer has been validated or not.
2	If client passed Postal Code then it will be passed to the acquirer as part of online authorisation
3	The client can control whether the Guardian software prompts for CVV entry on the ped via the promptForCvv parameter.

## ReportHook

```
Public Enum REPORTHOOK_REPORTTYPE
```

```
    REPORTHOOK_RT_ZREPORT = 0
```

```
    REPORTHOOK_RT_XREPORT
```

```
End Enum
```

```
Public Function Process(ByRef tillInformation As TillInformation,
```

```
    ByVal reportType As REPORTHOOK_REPORTTYPE) As Boolean
```

Method return TRUE on success, otherwise false on failure

	Notes
1	This TillInformation object will contain information specific to the site till which will be used for the header section of printed receipts
2	Report is printed by Guardian and not returned to the EPOS software.

## ReceiptProvider

```
Public Enum INTEGRAL_RECEIPTTYPE
    RECEIPTTYPE_UNKNOWN = 0
    RECEIPTTYPE_MERCHANTAUTHORISED
    RECEIPTTYPE_MERCHANTDECLINED
    RECEIPTTYPE_MERCHANTCANCELLED
    RECEIPTTYPE_CUSTOMERAUTHORISED
    RECEIPTTYPE_CUSTOMERDECLINED
    RECEIPTTYPE_CUSTOMERCANCELLED
End Enum

Public Function GetReceiptCount() As Integer
Public Function GetReceiptText(ByVal IReceiptIndex As Integer) As String
Public Function GetReceiptType(ByVal IReceiptIndex As Integer) As INTEGRAL_RECEIPTTYPE
Public Sub InitialiseReceipts()
Public Function IsSignatureRequired(ByVal IReceiptIndex As Integer) As Boolean
```

	Notes
1	This class is used to get receipt data dynamically while a transaction is progress. The EPOS software must support multi-threading to use this class.

## ReceiptProvider Sample Code

```
Imports Integral.Library.GuardianClient
Imports Integral.Library.GuardianClient.TransactionHook
Imports Integral.Library.GuardianClient.TokenTransactionHook

Private _receiptProvider As ReceiptProvider
Private _transactionComplete As Boolean = False

'Check if there are any receipts to print
Private Sub CheckForReceipts()

    Dim receiptCount As Integer = 0
    Dim counter As Integer = 1

    While Not _transactionComplete

        If receiptCount < _receiptProvider.GetReceiptCount() Then

            receiptCount = _receiptProvider.GetReceiptCount()

            While counter <= receiptCount

                MessageBox.Show(_receiptProvider.GetReceiptText(counter - 1))
                counter += 1

            End While

            System.Threading.Thread.Sleep(300)

        End If
```

```

    End While

End Sub

'Process transaction
Private Sub ProcessTransaction()

    Dim process As New TransactionHook
    Dim tillInfo As New TillInformation
    Dim transInfo As New TransactionInfo
    Dim receiptchecker As New System.Threading.Thread(AddressOf CheckForReceipts)

    _transactionComplete = False

    tillInfo.MerchantName = "TEST"
    tillInfo.Address1 = "ADDR1"
    tillInfo.Address2 = "ADDR2"

    'Create the module level receipt provider object
    _receiptProvider = New ReceiptProvider
    _receiptProvider.InitialiseReceipts()

    'Start the secondary thread to check for receipts to print
    receiptchecker.Start()

    process.Process(TRANSACTIONHOOK_TRANSACTIONTYPE.INT_TT_SALE, 100, tillInfo, transInfo)
    _transactionComplete = True

End Sub

```

## TaxfreeHook – Tax Free Shopping

Taxfree forms can be issued through Guardian for both card based sales and cash based sales. After a card based transaction is processed through Guardian, the TransactionInfo object will indicate if the card used is eligible for taxfree or not. The EPOS should prompt the operator to ask the customer if they want a taxfree form issued. This question should be asked also if paying by cash.



```
Public Enum TAXFREEHOOK_OPERATION
    INT_TT_ANYOPERATION = 0
    INT_TF_ISSUENEWFORM
    INT_TT_VOIDFORM
    INT_TT_REPRINTFORM
End Enum

Public Function Process(ByVal taxfreeOperation As TAXFREEHOOK_OPERATION,
    ByVal transactionAmount As Integer,
    ByVal taxfreeFormNumber As String,
    ByVal transactionReference As Integer,
    ByRef tillInfo As TillInformation,
    ByRef taxfreeInfo As TaxfreeInfo,
    ByVal customerReceiptDisabled As Boolean) As Boolean

Public Sub InitialiseProductInfo()
Public Sub AddProductInfo(ByVal productCode As String,
    ByVal amount As Integer,
    ByVal quantity As Integer)
```

	Description
1	If taxfree operation indicates 'Void Form' or 'Reprint Form' then the EPOS software should pass in the taxfree form number that was previously issued.
2	If issuing a new taxfree form then the EPOS software should use the AddProductInfo() method to add the products that are being sold as part of the transaction before calling the Process() method.
3	The allowed product codes will be provided by Sage Pay

## TaxfreeInfo

Parameter	Description
SaleAmount	Total Sale Amount
RefundAmount	Amount to be refunded to customer
VatRate	VatRate of product
AdministrationFee	Administration Fee charged by taxfree provider
MerchantReceipt	If transaction successful processed (i.e. no aborted by user) then this property will contain the merchant receipt text with carriage return / linefeed characters inserted as appropriate.
CustomerReceipt	If transaction successful processed (i.e. no aborted by user) then this property will contain the customer receipt text with carriage return / linefeed characters inserted as appropriate.
TaxfreeReceipt	If transaction successful processed (i.e. no aborted by user) then this property will contain the taxfree refund form with carriage return / linefeed characters inserted as appropriate.

## TaxfreeHook Code Sample

```
Imports Integral.Library.GuardianClient
```

```
Public Function IssueTaxfreeForm(ByVal transactionReference As Integer) As Boolean
```

```
    Dim returnValue As Boolean = False
    Dim taxfreeInformation As New TaxfreeInfo
    Dim tillInformation As New TillInformation
    Dim taxfreeTransaction As New TaxfreeHook
```

```
    'Add the products associated with the transaction
    taxfreeTransaction.InitialiseProductInfo()
    taxfreeTransaction.AddProductInfo("100", 10900, 1)
    taxfreeTransaction.AddProductInfo("104", 5000, 2)
```

```
    'Populate the till information object object
    tillInformation.MerchantName = "Test Shop"
    tillInformation.Address1 = "Test Address 1"
    tillInformation.Address2 = "Test Address 2"
    tillInformation.Address3 = "Test Address 3"
```

```
    'Issue (and print) the tax free form
    If taxfreeTransaction.Process(TaxfreeHook.TAXFREEHOOK_OPERATION.INT_TF_ISSUENEWFORM, _
        15900, _
        "", _
        transactionReference, _
        tillInformation, _
        taxfreeInformation, _
        False) Then
```

```
        'Taxfree form issued and printed successfully.
        'taxfreeInformation object will contain details including the issued form number
```

```
    Else
        'Failed processing request
    End If
```

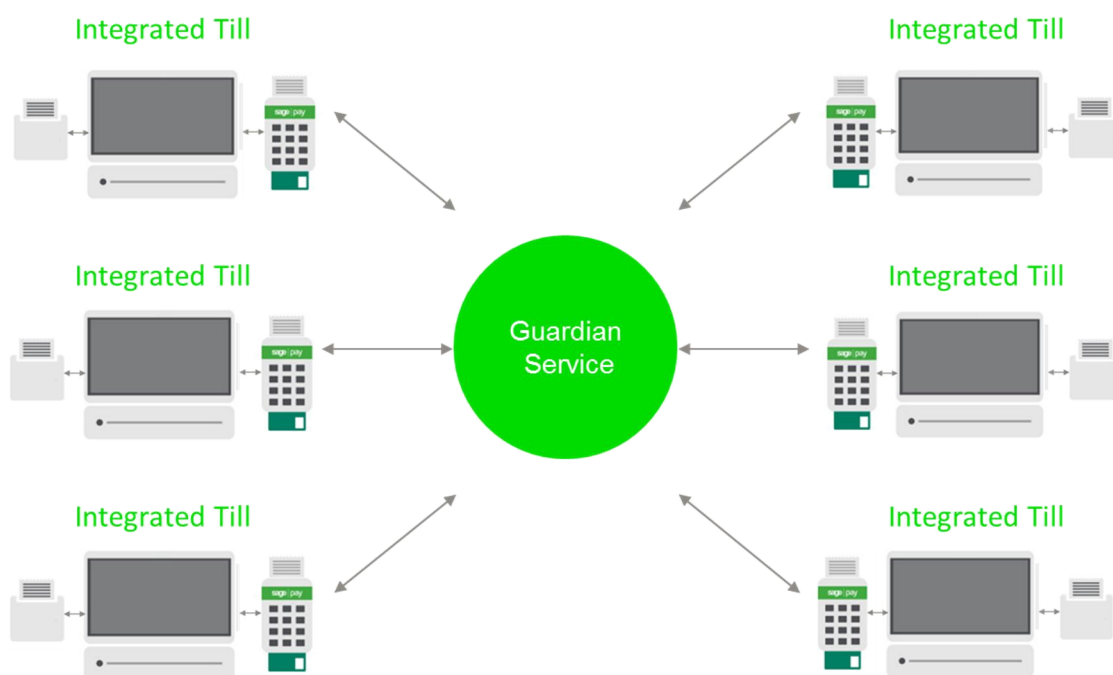
<p>Return returnValue</p> <p>End Function</p>
<pre> Public Function VoidTaxfreeForm(ByVal formNumber As String) As Boolean      Dim returnValue As Boolean = False     Dim taxfreeInformation As New TaxfreeInfo     Dim tillInformation As New TillInformation     Dim taxfreeTransaction As New TaxfreeHook      'Add the products associated with the transaction     taxfreeTransaction.InitialiseProductInfo()     taxfreeTransaction.AddProductInfo("100", 10900, 1)     taxfreeTransaction.AddProductInfo("104", 5000, 2)      'Populate the till information object object     tillInformation.MerchantName = "Test Shop"     tillInformation.Address1 = "Test Address 1"     tillInformation.Address2 = "Test Address 2"     tillInformation.Address3 = "Test Address 3"      'Void taxfree form specified by formNumber     If taxfreeTransaction.Process(TaxfreeHook.TAXFREEHOOK_OPERATION.INT_TT_VOIDFORM, _         15900, _         formNumber, _         0, _         tillInformation, _         taxfreeInformation, _         False) Then          'Taxfree form voided and printed successfully.         returnValue = True      Else         'Failed processing request     End If      Return returnValue  End Function </pre>

## IFSF

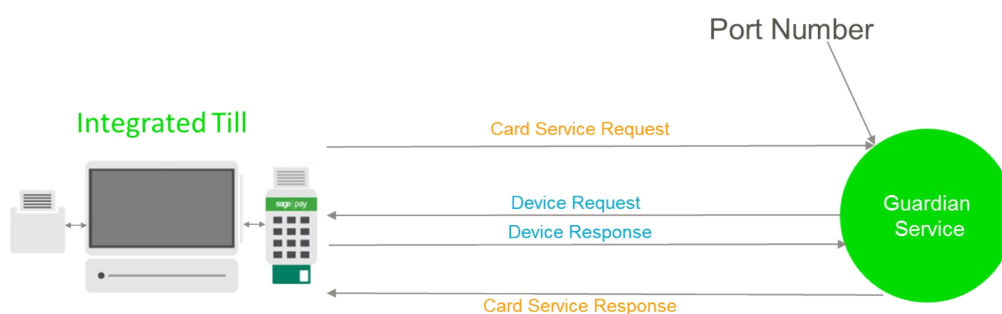
Sage Pay follow the International Forecourt Standards Forum (IFSF standards) which are not covered in this document. The full standards can be found at the following website.

<https://www.ifsf.org/documents>

In IFSF mode, Guardian runs as a service that is listening on a configured TCP/IP port. The following diagrams indicate at a high level the flow between the EPOS software and the Guardian service.



The following diagram shows the flow of the till communicating with Guardian



## Partial Fuel Authorisation

Guardian supports the concept of partial fuel authorisation. The POS system does not necessarily know what card type (credit,debit or fuel) the customer will present. Therefore for each transaction the POS send out a list of SagePay product codes (e.g. 002 (unleaded)) along with the value of this product in the current sale. If a fuel card is presented by the customer, then Guardian will validate that the products presented by the POS as part of the transaction are allowed for the fuel card that is presented.

### Partial Fuel Authorisation [OFF]

If partial preauthorisation is disabled then in the situation where some products presented by the POS are rejected by Guardian then the whole transaction fails and an error is returned to the POS. Guardian will return a colon separated list of disallowed fuel products via the TransactionInfo object. The POS system can choose to split tender the transaction and only send through the allowed products to Guardian or just go back to sub-total.

### Partial Fuel Authorisation [ON]

If partial preauthorisation is enabled then in the situation where some products presented by the POS are rejected by Guardian, Guardian will proceed with authorisation of the products that are allowed for the card. Guardian will return to the POS the outcome of the transaction (authorised or not) along with the reduced value of the transaction and also a colon separated list of disallowed fuel products via TransactionInfo object. The POS system can automatically split tender the transaction and look for payment for outstanding amount of transaction.

Partial Fuel Authorisation is enabled / disabled via the Server Configuration tab of the Configuration utility.

## Testing Utilities

### Guardian Tester

Integral.WinApp.GuardianTester.exe should be used to test connectivity initially.

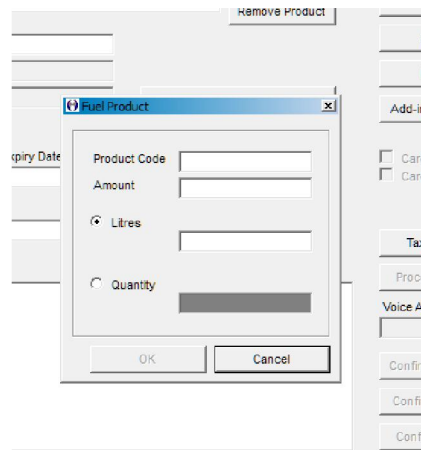
The screenshot shows the 'PayRouter Guardian Tester' application window. It features a 'Select Transaction Type' dropdown menu with 'SALE' selected. Below this is a 'Fuel Transaction Products' section with a text input field, 'Add Product', and 'Remove Product' buttons. Further down are input fields for 'Transaction Amount', 'Gratuity Amount', and 'Cashback Amount', along with a 'Calculate from Fuel Products' button. The 'Token' and 'Expiry Date' fields are empty, and the 'Transaction Mode' dropdown is set to 'Customer Not Present'. A 'Reference Text' input field is also present. On the right side, there is a vertical stack of buttons: 'Process Transaction', 'Void Transaction', 'Duplicate Receipt', 'X-Report', 'Z-Report', 'Run Task', 'Add-in Configuration', 'Taxfree Tester', 'Process (Non GUI)', 'Voice Auth Code' (with an input field), 'Confirm (Authorised)', 'Confirm (Cancelled)', and 'Confirm (Declined)'. At the bottom right is a 'Close' button. The bottom of the window has a 'Results' section with a large text area and a table with two columns: 'Receipt Type' and 'Receipt Text'.

Receipt Type	Receipt Text
--------------	--------------

It is designed to utilise the classes above to verify that transactions can be successfully processed.

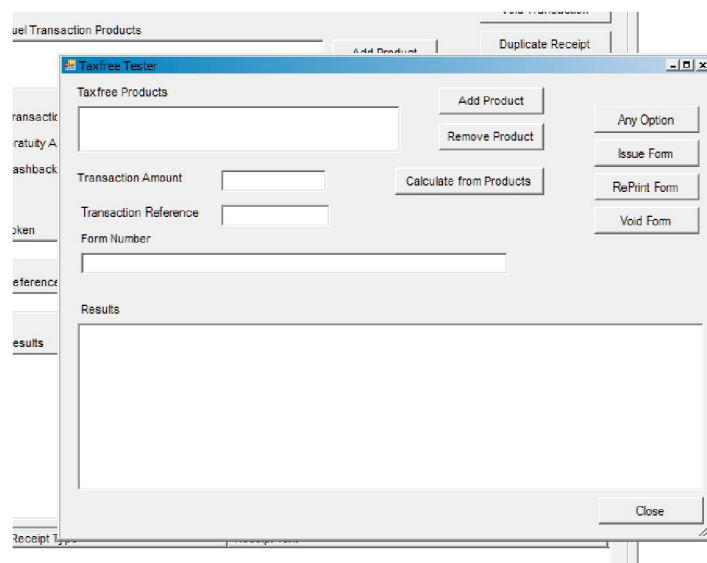
To test the simplest form of transaction, simply configure your environment to connect to a PED and receipt printer. Enter a transaction amount in the lowest denomination (100 for €/GBP 1, 2000 for 20.00 etc.). Click the process transaction button and the transaction will be processed.

You can also process using fuel products – click the Add Product button:

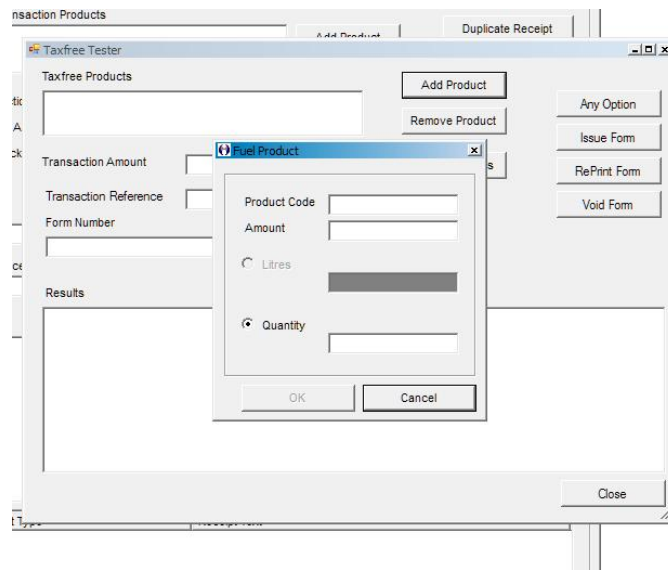


Follow the guidelines above for information on adding products.

This utility can also be used to test Taxfree transactions. Click the Taxfree tester button to load this screen:



To enter the Taxfree products (following the Taxfree product details above) click the Add Product button



When you have added all the products, click the Calculate from Products button and continue on to process the transaction.