September 2021

ELECTRONIC VOUCHER

For Electronic Cash Register

Edenred Belgium
Boulevard du Souverain-Vorstlaan, 165 bte 9
1160 Bruxelles / 1160 Brussel - BELGIUM
www.edenred.be

Contents

Abbreviation 3
Introduction 3
Description of the integration 4

VIC messages 5
- VMC_debit_request of Meal or Eco or other electronic voucher 5
- PDV_debit_result of Meal or Eco or other electronic voucher 6

Available configurations 8
- Mono-product contract (Meal or Eco or other voucher activated) 8
- Multi-product contract (Meal and Eco and/or other voucher activated) 8

Edenred Belgium
Boulevard du Souverain-Vorstlaan, 165 bte 9
1160 Bruxelles / 1160 Brussel - BELGIUM
www.edenred.be

Abbreviation

ECR	Electronic Cash Register
PDV	Purchase Device
VMC	Vending Machine Controller
V.I.A.	Voucher Issuers Association

V.I.A. Voucher Names	**Edenred Vouchers Names**
lunch-Voucher	Ticket Restaurant
eco-Voucher	Ticket EcoCheque
cadeau-Voucher	Ticket Compliments
sport culture-Voucher	Ticket Sport & Culture
consumption-Voucher	Consumption Voucher Edenred
book-Voucher	/
transport-Voucher	/

Introduction

The purpose of this document is to describe the recommended way of interfacing a cash register system with an attended terminal in order to perform a transaction with payment option selection of meal/eco/gift/sport-culture/consumption electronic vouchers with an Edenred card.

From a practical point of view, the UID/TID combination has to be activated by the issuer and this will allow test transactions. To obtain test cards the integrators can contact the issuer directly.

For more information about this situation, please contact Edenred on the following address: Paulo STEEGMANS, paulo.steegmans@edenred.com

Description of the integration

The principle of the solution consists of sending from the Electronic Cash Register (ECR) the *product code* (Meal/Eco/Gift/Sport-Culture/Consumption Voucher) that will be used for the transaction as payment means. This *product code* will be transmitted through the VIC protocol to the payment terminal. In turn, the terminal will transmit the *product code* to the Transaction Host Processor. Based on the product code, the Transaction Host Processor will do the necessary to identify and debit the corresponding account of the cardholder.

The field to be used through the VIC interface in order to start a transaction for a specific product is the vic_data (field number 36) in the vmc_debit_request message. Its content is composed of a set of data where each data owns a specific typed code. The type code that must be used in this context is the type code value 05 "discretionary data". Its value must respect the one defined as Product Category. From now on, the product code used in this context will be called an « ***Acquirer Discretionary Data*** » field value (***ADD***).

If ADD is provided, the host will check the presence and order of the ADD fields. The host will also check if the selected product is allowed in the merchant contract and will refuse the transaction if it is not the case. Otherwise the host should process the transaction based on the information provided in the ADD field. If ADD is not provided, or missing the multi-product tag, the host should process the transaction using the product-id derived from the merchant contract. In the case where the host receives a request without ADD and multi-product is linked to the account, the transaction will be refused.

VIC messages

How to modify the VIC protocol in your cash register application to be able to accept the Lunch Voucher, Eco Voucher, Gift Voucher, Sport-Culture Voucher and/or Consumption Voucher (or other electronic product in the future).

VMC_debit_request of Meal or Eco or Gift or other electronic voucher

FIELD		FORMAT		COMMENTS
index	**name**	**#bytes**	**type**	
-	vic_protocol_id	2	4x	Always **01 07**
-	vic_msg_code	2	2a	Always **56 44** (=VD)
-	vic_bit_map	24	192b	
3	vic_tx_amt	3	6x	Amount in eurocent (hexadecimal!). Example: 10,00 EUR -> **00 03 E8** Range: [**00 00 01..FF FF FF**]
12	tx_type	1	2i	Always **04** (single transaction)
16	vic_card_ind	1	2i	Always **01** (remove card after transaction)
20	vic_to	2	4x	Time given for card insertion. Always **00 1E** (30")
21	vic_tx_id	4	8x	Transaction identifier issued by VMC. Must be different for every transaction. Range : [**00 00 00 01..FF FF FF FF**]
23	Curcy	2	4i	Always **09 78** (EURO)
25	vic_cust_ind	1	2i	Always **01** (ask for cardholder validation)
36	**vic_data**		llllvar	All values for the discretionary_data (Type code value: 05): **200001# (lunch-Voucher)** **200002# (eco-Voucher)** **200003# (cadeau-Voucher)** **200004# (sport culture-Voucher)** *200005# (book-Voucher)* *200006# (transport-Voucher)* **200007# (consumption-Voucher)**
143	operator_nr	2	4i	Identifies the operator of the VMC. Range : [**00 01 .. 99 99**]
179	vic_version	1	2i	Always **13** (vic_protocol_subversion 1.07/13)

Edenred Belgium
Boulevard du Souverain-Vorstlaan, 165 bte 9
1160 Bruxelles / 1160 Brussel - BELGIUM
www.edenred.be

PDV_debit_result of Meal or Eco or Gift or other electronic voucher

FIELD		FORMAT		COMMENTS
index	**name**	**#bytes**	**length**	
-	vic_protocol_id	2	4i	Always **01 07**
-	vic_msg_code	2	2a	Always **50 44** (=PD)
-	vic_bit_map	24	192b	
2	term_id	8	8a	Terminal identifier
3	vic_tx_amt	3	6x	Transaction amount as asked by the VMC. Range: [**00 00 00** .. **FF FF FF**]
4	iep_tx_inc	2	4i	Indicates the result of the transaction range: [**00 00** .. **99 99**]
5	lg_cust	1	2i	Indicates the language of the card used range : [**00** .. **99**]
9	pdv_state	4	32b	Indicates the 'state' of the PDV. Irrelevant.
11	card_id_disp	24	24a	Contains the card nr, sent in "ready to print" ASCII-format; some digits of the number may be scrambled (depends on acquirer's security rules)
21	vic_tx_id	4	8x	Echo of transaction identifier issued by the VMC. Range: [**00 00 00 01** .. **FF FF FF FF**]
22	vic_msg_type	1	8b	Default **00**. If **10**, indicates that a message has to be displayed by the VMC, as given in the field display_message or referenced by the field display_text_ref (depends on the presence of the fields).
23	curcy	2	4i	Always **09 78** (EURO)
36	vic_data	2 + 1..999	4x + 1..199 8x	May contain different information from the acquirer. See full spec for format and details. e.g.: acquirer-issued authorisation code for the transaction and the used discretionary data of the debit_request message. **Example: 00170706363135373234080532303030303223**
125	<tx>vic_tx_amt	3	6x	Transaction amount as debited by PDV[1]. Range: [**00 00 01** .. **FF FF FF**]
126	<tx>curcy	2	4i	Always **09 78** (EURO)

Edenred Belgium
Boulevard du Souverain-Vorstlaan, 165 bte 9
1160 Bruxelles / 1160 Brussel - BELGIUM
www.edenred.be

145	ticket_data	2 + 1..999	4i + 1..999 a	The first two bytes announce the length of the field (only digits), **these two bytes not included**. Contains data (sent in "ready-to-print" ASCII-format) which **must** be printed by the VMC.
159	display_data	2 + 1..999	4i + 1..999 a	This field has to be taken into account only if indicated by the vic_msg_type field. The first two bytes announce the length of the field (only digits), **these two bytes not included**. Contains data (sent in "ready-to-print" ASCII-format) that **must** be displayed by the VMC.
170	transaction_protocol	2	4i	Info about communication between PDV and host. Range : [**00 00** .. **00 34**] (1st byte always **00**)
171	vic_bit_map_application_id	16	128b	Identifies the card application used for the transaction.
172	transaction_identifier	4	8i	Acquirer-issued identifier of the transaction. Range : [**00 00 00 00** .. **99 99 99 99**]
173	date_and_time	7	14i	Time of the transaction[2] Format : **YYYY MM DD hh mm ss**
174	brand_id	2	4i	Brand_id of the card used. Range : [**00 00** .. **99 99**]
176	cvm	2	4i	Info about how cardholder validates the transaction. range : [**00 00** .. **00 22**] (1st byte always **00**) Cardholder must sign the merchant copy of the transaction ticket only if cvm = xx**1**x.
178	brand_name	1 + 1..24	2i + 1..24a	The first byte indicates the length of the field, this byte not included. The following bytes contain the brand name. This field identifies the means of payment used by the cardholder.
179	vic_version	1	2i	Always **13** (vic_protocol_subversion 1.07/13)
182	display_text_ref	2	4x	This field has to be taken into account only if indicated by the vic_msg_type field. Contains a reference to a message to be displayed by the VMC.

[1] In the case of Partial Approval (indicated by an iep_tx_inc at **00 01**), this amount can be different than the one specified in the field vic_tx_amt. Partial Approval is only available in VIC 1.07/13 (a re-certification is required to support P.A.)

[2] As logged by the acquirer, or by the terminal in case of emergency fall-back

Available configurations

Mono-product contract (Meal or Eco or Gift or Sport-Culture or Consumption Voucher contract activated)

Accepted:

- Transaction Processing Host receives request without ADD multi-product tag 20 (product selection is done based on contract)
- Transaction Processing host receives request with ADD multi-product tag 20 and selected product is allowed in the contract (well-formatted and correct value)

Refused:

- Transaction Processing host receives request with ADD multi-product tag 20 but not well-formatted and/or with an incorrect value
- Transaction Processing host receives request with ADD multi-product tag 20 and selected product is not allowed in contract

Multi-product contract (Meal and Eco and/or Gift Voucher and/or Sport-Culture and/or Consumption Voucher contract activated)

Accepted:

- Transaction Processing Host receives request with ADD multi-product tag 20 (well-formatted and correct value)

Refused:

- Transaction Processing Host receives request with ADD multi-product tag 20 not well-formatted and/or with an incorrect value
- Transaction Processing Host receives request without ADD multi-product tag 20