

## Electronic Voucher

---

V1.1

## References

[1] Reference Functional Specifications VIC Standard

## Abbreviation

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

## 1. Description of the integration

The principle of the solution consists of sending from the Electronic Cash Register (ECR) the *product code* (Eco 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.

Contact person for technical questions: [kenzai.abderrahmane@sodexo.com](mailto:kenzai.abderrahmane@sodexo.com)  
+32 484 71 82 77

### 3. VIC messages

How to modify the VIC protocol in your cash register application to be able to accept the Lunch Voucher and/or Eco Voucher (or other electronic product in the future)

#### 3.1. VMC\_debit\_request of electronic voucher

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

#### 3.2. PDV\_debit\_result of Meal or Eco voucher

FIELD		FORMAT		COMMENTS
index	name	# bytes	length	
-	vic_protocol_id	2	4i	Always <b>01 07</b>
-	vic_msg_code	2	2a	Always <b>50 44</b> (=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: <b>[00 00 00 .. FF FF FF]</b>
4	iep_tx_inc	2	4i	Indicates the result of the transaction range: <b>[00 00 .. 99 99]</b>
5	lg_cust	1	2i	Indicates the language of the card used range : <b>[00 .. 99]</b>
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..1998x	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. <b>Example: 00170706363135373234080532303030303223</b>
125	<tx>vic_tx_amt	3	6x	Transaction amount as debited by PDV <sup>1</sup> . Range: [00 00 01 .. FF FF FF]
126	<tx>curcy	2	4i	Always 09 78 (EURO)
145	ticket_data	2 + 1..999	4i + 1..999a	The first two bytes announce the length of the field (only digits), <b>these two bytes not included</b> . Contains data (sent in "ready-to-print" ASCII-format) which <b>must</b> be printed by the VMC.
159	display_data	2 + 1..999	4i + 1..999a	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), <b>these two bytes not included</b> . Contains data (sent in "ready-to-print" ASCII-format) that <b>must</b> be displayed by the VMC.
170	transaction_protocol	2	4i	Info about communication between PDV and host. Range : [00 00 .. 00 34] (1 <sup>st</sup> 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 <sup>2</sup> 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] (1 <sup>st</sup> byte always 00) Cardholder must sign the merchant copy of the transaction ticket only if cvm = xx1x.
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.

See **Annex 1: VIC logging of an electronic Voucher transaction**

<sup>1</sup> 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.)

<sup>2</sup> As logged by the acquirer, or by the terminal in case of emergency fall-back

## 4. Available configurations

### 4.1. Mono-product contract

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

### 4.2. Multi-product contract

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

This integration is supported by all the VIC 1.07/xx versions. No additional VIC re-certification is required for this integration. These electronic voucher cards are supported by the attended terminals XENTA and YOMANI.

For technical support about the VIC integration, please contact the ECR-Vending certifications mailbox: [ecrvendingcertifications-belgium@worldline.com](mailto:ecrvendingcertifications-belgium@worldline.com)

### Annex 1

Event Number	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	ASCII
1	02	05	01	07	56	44	a0	11	1a	80	10	10	00	00	00	00	00
17	00	00	00	00	00	00	00	00	10	03	00	00	00	00	00	00	00
33	00	00	10	03	08	00	00	1e	f1	d3	b5	11	09	78	00	00	00
49	09	08	05	32	30	30	30	30	32	23	12	34	20	20	13	03	00
65	75	d6	06	05	02	1a	01	07	50	44	f8	a0	0e	00	10	10	00
81	00	00	00	80	00	00	00	00	00	00	00	00	0c	00	00	80	00
97	02	7d	60	00	30	31	36	36	33	30	31	34	00	00	10	03	00
113	00	00	04	40	00	00	00	78	78	78	78	78	78	78	78	78	00
129	78	78	78	36	34	37	32	20	20	20	20	20	20	20	20	f1	00
145	d3	b5	11	00	09	78	00	17	07	06	36	31	35	37	32	34	00
161	08	05	32	30	30	30	30	32	23	00	00	10	03	09	78	07	00
177	19	43	4c	49	45	4e	54	20	54	49	43	4b	45	54	0d	0a	00
193	2d	2d	2d	2d	2d	2d	2d	2d	2d	2d	2d	2d	2d	2d	2d	2d	00
209	2d	2d	2d	2d	2d	2d	2d	2d	2d	2d	2d	2d	2d	2d	2d	2d	00
225	2d	2d	2d	2d	2d	2d	0d	0a	04	0e	20	20	20	20	20	20	00
241	20	20	20	20	20	20	20	20	20	20	20	20	20	0d	0a	04	00
257	0e	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	00
273	20	20	20	20	0d	0a	2d	2d	2d	2d	2d	2d	2d	2d	2d	2d	00
289	2d	2d	2d	2d	2d	2d	2d	2d	2d	2d	2d	2d	2d	2d	2d	2d	00
305	2d	2d	2d	2d	2d	2d	2d	2d	2d	2d	2d	2d	0d	0a	04	54	00
321	65	72	6d	69	6e	61	6c	3a	20	30	31	36	36	33	30	31	00
337	34	0d	0a	04	4d	65	72	63	68	61	6e	74	3a	20	30	30	00
353	30	30	30	31	30	34	20	20	20	20	20	20	20	0d	0a	04	00
369	50	65	72	69	6f	64	3a	20	30	30	33	31	20	20	20	20	00
385	20	54	72	61	6e	73	61	63	74	69	6f	6e	3a	20	30	30	00
401	30	30	30	31	34	35	0d	0a	0d	0a	04	0e	20	20	20	53	00
417	6f	64	65	78	6f	20	65	50	61	73	73	20	20	20	20	0d	00
433	0a	0d	0a	20	20	20	20	20	20	20	20	20	20	20	20	28	00
449	41	30	30	30	30	30	30	32	36	36	30	31	29	20	20	20	00
465	20	20	20	20	20	20	20	20	20	0d	0a	0d	0a	04	0e	43	00
481	61	72	64	3a	20	78	78	78	78	78	78	78	78	78	78	78	00
497	78	36	34	37	32	20	20	0d	0a	0d	0a	03	61	72	64	00	00
513	20	73	65	71	75	65	6e	63	65	20	4e	72	3a	20	31	0d	00
529	0a	56	61	6c	69	64	20	74	68	72	75	3a	20	33	30	2f	00
545	30	34	2f	32	30	0d	0a	0d	0a	54	45	53	54	20	45	56	00
561	43	33	33	31	0d	0a	0d	0a	04	50	41	59	4d	45	4e	54	00
577	20	20	20	20	20	20	20	20	20	20	20	20	0d	0a	04	44	00
593	61	74	65	3a	20	32	31	2f	30	31	2f	32	30	31	36	20	00
609	31	34	3a	32	34	20	20	20	20	20	20	20	20	20	20	20	00
625	20	20	20	20	20	0d	0a	0d	0a	04	41	75	74	68	2e	20	00
641	63	6f	64	65	3a	20	36	31	35	37	32	34	20	20	0d	0a	00
657	0d	0a	0e	22	53	6f	64	65	78	6f	20	43	61	72	64	20	00
673	2f	20	45	63	6f	20	50	61	73	73	22	0d	0a	0d	0a	0d	00
689	0a	04	0e	54	6f	74	61	6c	3a	20	20	20	20	20	30	2c	00
705	30	33	20	45	55	52	0d	0a	0d	0a	52	65	61	64	2d	6d	00
721	6f	64	65	3a	20	20	20	43	68	69	70	0d	0a	0e	0e	44	00
737	69	73	70	6f	6e	69	62	6c	65	20	31	34	39	2c	38	34	00
753	20	45	55	52	0d	0a	0e	64	6f	6e	74	20	31	34	39	2c	00
769	38	34	20	45	55	52	0d	0a	0e	65	78	70	69	72	65	20	00
785	6c	65	20	31	37	2f	30	31	2f	32	30	31	37	0d	0a	20	00
801	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	00
817	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	00
833	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	00
849	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	00
865	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	00
881	20	20	20	20	20	20	20	20	20	20	20	20	20	0d	0a	0d	00
897	00	00	62	00	00	12	00	00	00	00	00	00	00	00	04	00	00
913	00	00	00	00	00	00	00	00	01	45	20	16	01	21	14	24	00
929	47	30	10	03	00	22	12	53	6f	64	65	78	6f	20	65	50	00
945	61	73	73	13	03	c2	d6	06	1a	@							00