



Rowa® Technologies

Becton Dickinson Rowa Germany GmbH

WWKS 2+ for Self-Checkout

Revision 0.6.1

Revisions

Revisions	Date	Change	Author
0.1	20.08.2020	Self-Checkout related content extracted from full WWKS 2+ spec v0.26	Andreas Kohlhof
0.2	28.08.2020	Minor fixes in payment examples	Andreas Kohlhof
0.3	08.02.2021	Member price category added. ViewPointId in shopping cart deprecated. New capability ShoppingCartArticleMaster for transmitting article master data embedded in ShoppingCart	Andreas Kohlhof
0.4	02.03.2021	Introduction of update elements in the ShoppingCartUpdateRequest that clearly indicate the requested update. IT system may just use these and fully ignore the shopping cart transmitted by device in the update request. FormattedString data type for description attributes that contain basic display formatting information. Added Description attribute to ShoppingCartItem. Added ItemListPrice and ItemOfferPrice to ShoppingCartItem. Display languages in shopping carts. PIS may ask device user a question when handling a ShoppingCartUpdateRequest.	Andreas Kohlhof

0.5	19.03.2021	<p>Removed <code>ShoppingCart</code> from <code>ShoppingCartUpdateRequest</code> samples to make clear that they are for information purpose only.</p> <p><code>ShoppingCartId</code> now attribute of root element of <code>ShoppingCartUpdateRequest</code>.</p> <p>Introduced dedicated capability <code>PartialPayment</code> for partial payment and payment with bonus points.</p> <p>Introduced <code>ReceiptWidth</code> attribute in <code>PaymentRequest</code> update element of <code>ShoppingCartUpdateRequest</code>.</p> <p>Allowed multiple <code>Receipt</code> elements within one <code>PaymentItem</code> to support receipt printing in several steps.</p> <p>Added <code>Id</code> attribute to <code>ShoppingCartUpdateRequest.Query</code> element.</p> <p>Introduced <code>Close</code> update element in <code>ShoppingCartUpdateRequest</code>. Finish with <code>SalesPointId=""</code> is now deprecated usage for indicating that device has finished handling of a shopping cart.</p> <p>Several detail clarifications.</p>	Frank Fröse, Andreas Kohlhof
0.5.3	05.05.2021	<p>Several minor error corrections.</p> <p>Added <code>ReceiptWidth</code> as attribute to <code>ShoppingCartUpdateRequest.Payment</code> element.</p> <p>Added additional receipt format "TEXT".</p>	Andreas Kohlhof
0.6	21.06.2021	Added <code>LastOrderedQuantity</code> to <code>ShoppingCartUpdateRequest.ChangeQuantity</code> element	Andreas Kohlhof
0.6.1	01.07.2021	Fixed position of <code>CustomerId</code> within <code>ShoppingCartUpdateResponse</code> in Scan of a loyalty card example (section 7.9.3 Scan barcode for shopping cart).	Andreas Kohlhof

Content

1	Introduction	7
2	Communication between a pharmacy IT system and connected devices.....	8
	2.1 Storage (Automated Storage and Retrieval Systems)	8
	2.2 Digital Shopping (Sales Support Devices)	8
	2.3 Commercial Excellence	8
3	Protocol design principles.....	9
	3.1 Assumptions	9
	3.2 Principles.....	10
	3.2.1 Family of independent device category protocols	10
	3.2.2 Public release of extension.....	10
	3.2.3 Version control	10
	3.2.4 No room for interpretation	11
	3.2.5 Open-closed principle for enhancements.....	11
	3.2.6 Protocol transparent connection	11
	3.2.7 Version configuration instead of dynamic negotiation	12
	3.2.8 Handling of required licenses for transmitted data outside the protocol	12
4	Technical basics of communication	13
5	Devices and components in the pharmacy.....	14
	5.1 Pharmacy workflows	14
	5.1.2 Basic workflow sale.....	14
	5.2 Point of sales.....	16
	5.3 Device numbers.....	17
	5.4 Device categories	18
	5.4.1 Pharmacy IT system	18
	5.4.3 Sales supporting devices at a point of sale.....	18
	5.4.3.4 Self-checkout cash register	19
6	Capabilities	21
	6.1 Connection setup (Connect)	21
	6.4 Item display.....	21
	6.4.1 Display item information (ItemDisplay)	21

6.5	Sales transactions.....	22
6.5.2	Manage shopping carts (ShoppingCarts).....	22
6.5.3	Scan barcodes (BarcodeScan)	23
6.5.5	Payment of shopping carts.....	23
6.5.5.1	Card terminal handling by Device (Payment).....	23
6.5.5.2	Card terminal handling by PIS (PISCardTerminal)	24
6.5.5.3	Receipt layout by PIS (ReceiptLayout)	24
6.5.5.4	Partial payment / payment with bonus points (PartialPayment)	25
6.5.6	Transfer of shopping carts (ShoppingCartTransfer)	25
6.5.8	ShoppingCart article master data (ShoppingCartArticleMaster)	26
7	Use cases	27
7.1	Connection establishment	27
7.4	Prices	31
7.9	Sales transactions.....	33
7.9.1	Request shopping cart.....	34
7.9.2	Modify shopping cart.....	36
7.9.3	Scan barcode for shopping cart	42
7.9.5	Terminate handling of shopping cart at a device	49
7.9.6	Pay shopping cart.....	51
7.9.6.1	Card terminal handling by device	51
7.9.6.2	Card terminal handling by PIS	54
7.9.6.3	Receipt layout by PIS	56
7.9.6.4	Partial payment and payment with bonus points	58
7.9.12	Shopping cart with item master data.....	61
8	Data structures.....	63
8.1	Data types	63
8.2	Stock	65
8.2.1	Article	65
8.2.1.2	Prices	66
8.2.1.3	Tags	68
8.4	Sales Transaction (Shopping cart).....	69
8.11	System component (Subscriber)	78
9	Message reference I – General messages.....	80
9.1	Message structure.....	80

9.2 Initialization	81
9.2.1 HelloRequest	82
9.2.2 HelloResponse	85
9.3 Keep alive	87
9.3.1 KeepAliveRequest	88
9.3.2 KeepAliveResponse	89
9.4 Unknown messages	90
9.4.1 UnprocessedMessage	91
12 Message reference IV – Sales transactions	94
12.1 Request shopping cart/sales transaction	94
12.1.1 ShoppingCartRequest	95
12.1.2 ShoppingCartResponse	97
12.2 Modify shopping cart/sales transaction	99
12.2.1 ShoppingCartUpdateRequest	100
12.2.2 ShoppingCartUpdateResponse	109

1 Introduction

What is this document about?

This document describes communication between a pharmacy IT system and a device supporting pharmacy workflows via the WWKS 2+ protocol.

Messages are exchanged in both directions. Depending on the device and workflow, different messages of this specification are used.

This specification lists all available messages, their use and syntax. A software developer may use this information to enable successful communication between a pharmacy IT system and a device via WWKS 2+.

Who should read this document?

This document is intended for:

- Personnel involved in integrating a pick-up terminal to a pharmacy IT system
- Developers developing software (such as parsers) to translate between WWKS 2 messages and the pharmacy IT system

Definition of terms

The systems are abbreviated as follows in places within the text:

Abbreviation	Definition
PIS	Pharmacy IT system
SSD	Sales support device
ASRS	Automated storage and retrieval system. In the context of this document, an automated storage and retrieval system also means a system comprising multiple interconnected automatic storage machines with or without a shared control computer
FMD	Falsified Medicine Directive. EU regulation that enforces datamatrix codes on Rx packs and introduce serial number that allow to track packs.

2 Communication between a pharmacy IT system and connected devices

Although there are a lot of different devices and use cases, the data exchanged in a pharmacy is usually the same with just minor deviations. Therefore WWKS 2+ does not define device interfaces but models reoccurring workflows consistently to allow definition of new use cases using existing interface capabilities.

Depending on use case and devices some protocol elements marked as optional may become mandatory for this use case. Please refer to the description of the use cases for details.

The WWKS 2+ protocol defines the term **Capability** that allows PIS and device to signal their support for a specific use case and the protocol elements needed for it.

PIS and device may decide themselves, which capabilities they support. Anyway, if a capability is supported, all aspects of it have to be implemented.

For already known device types, chapter 0 lists the capabilities usually supported.

WWKS 2+ currently defines three main layers of supported systems:

2.1 Storage (Automated Storage and Retrieval Systems)

Automated Storage and Retrieval Systems (ASRS) have been supported already by previous versions of WWKS. WWKS 2+ is based on these interfaces and includes all well-known use cases and the messages defined in WWKS 2.

2.2 Digital Shopping (Sales Support Devices)

WWKS 2+ extends WWKS 2 by the support of additional use cases beyond the integration of ASRS', most of all the integration of devices supporting sales transactions in a pharmacy.

A Sales Supporting Device (SSD) enables pharmacy staff and customers to view or modify the state of a sales transaction.

Samples:

- Information displays
- Digital shelf
- Consultation terminals
- Self-service terminal
- Pick-up terminal
- Self-service cash register

2.3 Commercial Excellence

Next to devices that directly support logistic or sales workflows, WWKS 2+ allows to integrate software components that optimize the workflows and marketing activities managed by the pharmacy IT system.

These components access, modify and play back data managed by the PIS.

Examples:

- Optimized price calculation
- Optimizing of cross selling recommendations
- Category management

3 Protocol design principles

WWKS 2+ defines an open standard of communication protocols for the integration between a pharmacy IT system and supporting devices of different categories and vendors.

The protocol design bases on several assumptions about the use context and principles derived from the assumptions described in this chapter. This serves on the one hand for a better understanding and on the other hand represents a guideline for future protocol extensions.

3.1 Assumptions

The following assumptions form the basis of the protocol:

1. All system vendors in the pharmacy environment have interest in making their systems to cooperate as seamlessly as possible with other market players – for the benefit of their joined customers.
2. All system vendors want to minimize the number of interfaces they have to implement and want to reduce integration efforts as much as possible.
3. Communication protocols define both messages and behaviors. The more defined the behavior is, the less misunderstandings between communication partners may occur.
4. To minimize development efforts and error rates, the published protocol has to be stable over time.
5. Usage of well-established technologies like XML, XSD, JSON or REST supports reduction of development efforts and error rates.
6. Different systems of the same device category do not differentiate themselves in competition through the use of proprietary communication protocols, but in the functions that are based on the communication protocols. Communication protocols in real-life use should therefore be published at a central location in the sense of an open standard and be freely accessible and usable for all market partners without patent or license obligations.
7. Vendor specific enhancements of the communication protocols are unavoidable.
8. Within one pharmacy, systems of different vendors and different device categories might be in use. Protocol updates and enhancements for one area must not enforce updates in other areas.
9. The heterogeneous system landscape in the pharmacy does not permit to update all systems in parallel to a new version of the protocol. It must always be assumed that the different systems receive their updates according to their own rhythm.
10. The market acts faster than standardization bodies. New systems and new ideas often require new communication protocols. New protocols or protocol extensions should therefore be expediently promoted by the market partners involved. Central bodies are suitable for reaching agreements that an already published protocol is to be supported by

the association participants. For the development of completely new protocols, the central standardization bodies are usually too slow or the voting effort is unacceptably high.

11. WWKS2+ deals exclusively with communication between systems within the local, protected pharmacy network. Connection to systems which are operated outside the local pharmacy network requires further concepts, e.g. in the area of authentication and authorization, which are deliberately not dealt with in WWKS2+.

3.2 Principles

The above-mentioned assumptions result in principles that are applied in the definition, further development and expansion of WWKS2+.

3.2.1 Family of independent device category protocols

WWKS 2+ is a protocol family that consists of multiple dedicated device category protocols on the application layer. Each device category defines its own workflows based on a common WWKS 2+ base architecture. There must not be protocol dependencies between different device category protocols. The protocol of one device category must be able to advance without impact onto other categories.

3.2.2 Public release of extension

Market players who want to extend the published protocol to enable new features, may implement these themselves or ask the creator of the existing functionality. In complex cases consultation with other users may help to get broader support for the enhancement.

In any case, the enhancement should be made available to the public latest after successful test in real life. This avoids multiple different protocols enhancements for the same use case.

3.2.3 Version control

Each capability (see chapter 6) of WWKS 2+ and each future enhancement have an unambiguous version number following this schema:

```
WWKS2 <Capability>-<Major>.<Minor>[-<Variant>]
```

With

<Capability>:	Name of the capability e.g. PickingRobot ItemDisplay
<Major>:	Increasing number. Increased if the new revision contains breaking changes in the interface.
<Minor>:	Increasing number. Increased, if all changes of the new revision do not break existing implementations.
<Variant>:	Vendor defined.

Remark: Even if two market player agree to not disclose a protocol enhancement, they should use a unique version number to mark their enhancement.

Each new version contains

- A full textual description
- A description of changes to the previous version

- An XSD-Schema¹ and
- If possible, a programming library or a reference implementation for both sides of the communication channel.

3.2.4 No room for interpretation

The named version of a device protocol is a binding contractual basis between the communication partners. Each protocol version must be defined and described in such a way that it leaves no room for interpretation in the use and meaning of the defined messages and structures. A communication partner must always be able to rely on the fact that all market participants behave completely the same and version-conformant with the same protocol version.

3.2.5 Open-closed principle for enhancements

The WWKS2+ protocol architecture supports the open-closed principle². This means that each vendor supporting a device category protocol version assures to

- (a) Support behavior, structure and interpretation of messages and their content without changes permanently over time according to the protocol specification. It may expect this as well from its communication partners (closed protocol).

On the other hand

- (b) Both partners are allowed to transmit any additional messages or message attributes that are not part of the well-defined protocol version. The communication partner must be able to safely ignore unknown messages and message elements without system error or stop of communication (Open for protocol enhancements).

It is most important that an enhancement of the protocol never implies a structural or behavioral change of the existing protocol.

3.2.6 Protocol transparent connection

The establishment of the (network) connection needed to exchange the messages is not part of the device class protocol itself. Device class protocols act on the application layer but not on the session layer.

Usually the protocol client establishes the connection, though is not part of the protocol specification. Establishing and terminating the connection are protocol-transparent operations that must not be used to transmit application-specific states or information.

The start of a new session is initiated by a `HelloRequest/HelloResponse` handshaking, the termination of a session by a corresponding status message or a `new HelloRequest/HelloResponse` handshaking. In particular, a device class protocol must not stipulate that a communication partner must terminate the underlying connection in certain domain-oriented states. Every communication partner must expect the underlying connection to be interrupted at all times. The communication partner who previously initiated the connection setup is then responsible for re-establishing the connection.

¹ If necessary, the schema should use the `<openContent>` element of XSD 1.1 to enable extensions beyond the defined version in schema validations.

² https://en.wikipedia.org/wiki/Open%E2%80%93closed_principle

3.2.7 Version configuration instead of dynamic negotiation

The agreement of the concrete protocol version used between two communication partners is not part of WWKS2+ and takes place outside the protocol. Dynamic, program-controlled negotiation of the version is not provided for in WWKS2+. The version identifiers transmitted in HelloRequest and HelloResponse serve the sole purpose of logging and as an aid in the analysis of communication problems.

3.2.8 Handling of required licenses for transmitted data outside the protocol

Licenses may be required to use data transmitted via WWKS2+. The WWKS2+ protocol does not provide a check whether corresponding usage rights exist. This must be ensured outside the protocol.

4 Technical basics of communication

Communication between the PIS and the device is handled via the TCP/IP protocol.

Alternative transmission paths for the messages defined in this document are generally possible, but are not the subject of this specification.

The IP address of the device in the pharmacy network is either assigned via DHCP or configured as a fixed address.

Whether the device or the PIS acts as a server depends on the device class.

Device which are critical for pharmacy workflows (e.g. ASRS) or devices to which a PIS wants to establish multiple connections, usually acts as server. Device which are in case of failure non-critical for the pharmacy workflows (e.g. screens) usually act as clients. PIS and device vendor may agree on different setups.

The default port for communication via WWKS 2 is 6050. Different port configurations should be configurable on both sides.

It is possible to connect several device to the PIS via only one socket connection. To distinguish different devices, the messages contain device ids that are unique within the pharmacy. See sections 5.3 Device Numbers and 7.1 Connection establishment for details.

It is recommend to use different connections for different device categories.

Methods for securing the connection in unsafe environments are currently not part of the WWKS 2+ specification.

The respective server waits for incoming requests from the client. The data connection is established exclusively by the client and remains active as long as the device and PIS are in operation. The connection is not opened and closed for every single operation. The client is responsible for re-establishing the connection immediately after an abort. If this fails, it will repeat the recovery attempt at regular intervals.

UTF8-encoded XML messages are exchanged in both directions via the connection.

After establishing the connection, the client sends a `HelloRequest` within five seconds. The server closes the connection if no `HelloRequest` is received within this time period (see section 7.1 Connection establishment).

5 Devices and components in the pharmacy

With the WWKS2+ interfaces, a wide variety of devices and software components can be integrated into a pharmacy's sales and logistics processes managed by the pharmacy IT system.

The devices take over certain roles for which only parts of WWKS2+ are required. WWKS2+ is represented by capabilities that bundle related use cases in a meaningful way.

Devices and pharmacy IT systems implement a capability always completely and can rely on each other that the other side implements all use cases belonging to a capability. In the connection setup, both sides specify which capabilities are supported in the current configuration.

This chapter describes already known device categories and defines how they are integrated into the workflows of the pharmacy and which capabilities are required as mandatory or optional for PIS and device.

5.1 Pharmacy workflows

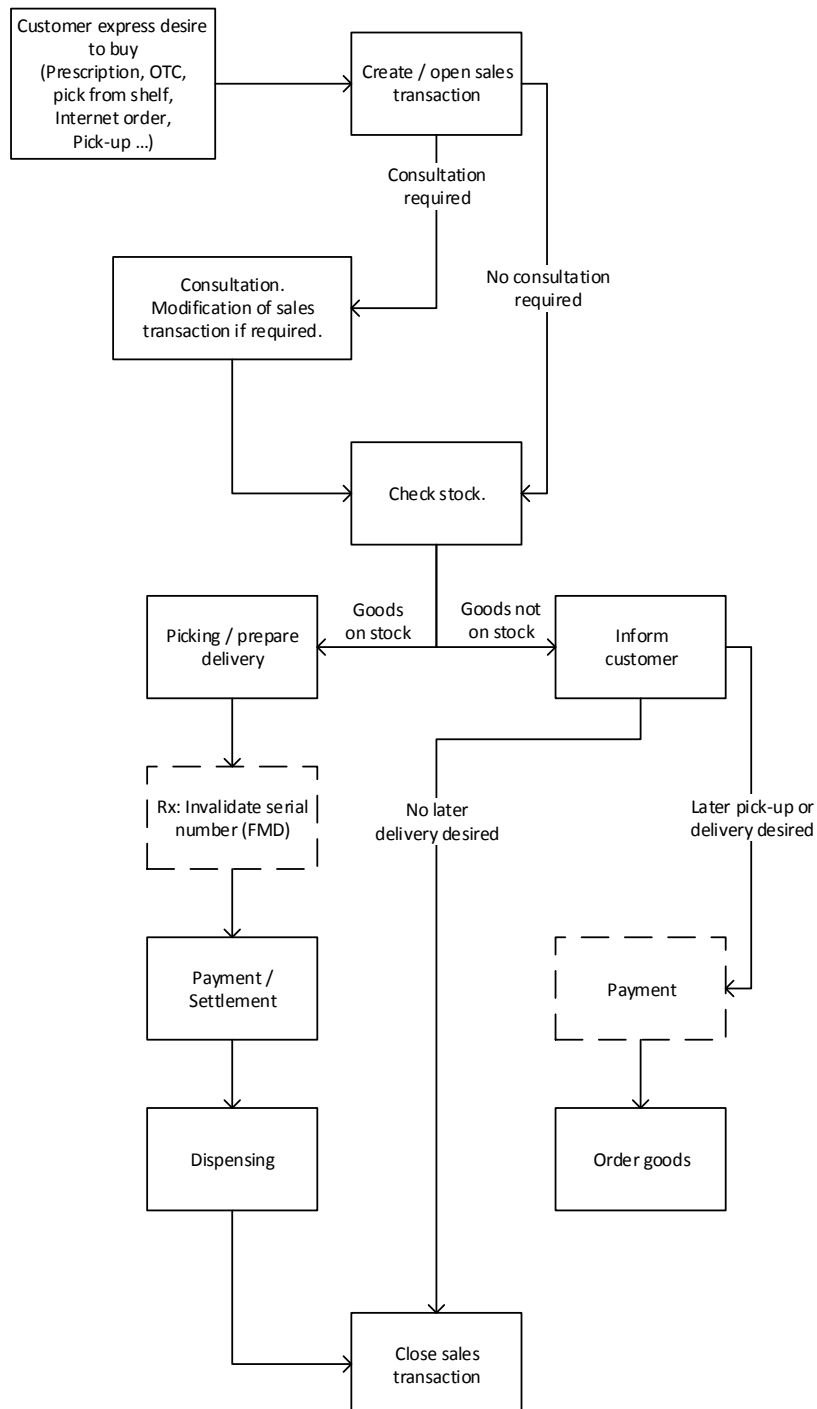
5.1.1 Basic workflow good receipt

Please refer to full WWKS2+ specification, not relevant for self-checkout terminals.

5.1.2 Basic workflow sale

Purpose

Customer buys items in the pharmacy, either in person or electronically

Sequence

5.2 Point of sales

A **point of sale** is a place in a pharmacy where only one customer can be served exclusively at a time. At a point of sale, several devices can participate together in the sales process for this customer.

A point of sale has an ID defined by the PIS. These are independent of the device numbers and can overlap with the number ranges for device numbers. For example, they can correspond to the device numbers of the PIS cash registers.



5.3 Device numbers

For communication via WWKS2+, each device that is connected to the pharmacy IT system (PIS) is uniquely identified by a device number.

The device number is configured in the device and communicated to the PIS during connection setup. The device manufacturer ensures that no two devices connected to the PIS via the same connection use the same number.

In each message and each request for a new use case, the initiator of the communication specifies its own device number as the `source` and the target device number as `destination`. In replies to these messages, the received `source` is returned as `destination` so that the receiver can route the response to the original message source. The own device number is entered as `source` in the reply.

Sub-components of a device can use their own device numbers in outgoing messages. They do not have to be listed explicitly in the `HelloRequest`, but then cannot be explicit recipients of messages from the PIS (except for replies to requests from the sub-component).

If a message is not addressed to a specific device, `destination "0"` is used.

The following device numbers and number ranges are defined:

Number	Device
0	Broadcast to all connected devices
100	PIS
101..199	Device of the PIS vendors (e.g. cash register).
200..999	Devices of other vendors connected to the PIS

5.4 Device categories

5.4.1 Pharmacy IT system

The pharmacy IT system (PIS) manages the stock of the pharmacy and has sovereignty over the sales and logistics processes in the pharmacy. It manages the article master and the stock.

It has at least one central server that manages the connections to the devices. In addition, there may be other PIS devices or components in the pharmacy which, if required, also establish direct connections to individual devices (in particular picking robots).

`Subscriber.Type` in the `HelloRequest`: „WMS“

5.4.2 Automated Storage and Retrieval System

Please refer to full WWKS2+ specification.

5.4.3 Sales supporting devices at a point of sale

A **sales supporting device** (SSD) is a device that is integrated temporarily in a sales workflow managed by the PIS. It is always assigned to a point of sale known to the PIS. A **point of sale** is a location in the pharmacy where, at one point in time, one patient is served exclusively.

All sales supporting devices assigned to a point of sale work together on the current **sales transaction** and may initiate modifications and actions (e.g. payment, delivery) on the transaction.

In case the sales transaction workflow continues at another point of sale, the PIS has to transfer the transaction to this point of sale explicitly.

The integration of sales systems that manage stock and transactions on their own is not in scope von WWKS2 2+.

A sales supporting device may be a device of the PIS vendor or of another vendor.

A SSD may be configured to support multiple points of sale, but must then ensure that it uniquely handles only one assigned point of sale at a time and supports the operator in switching between the points of sale.

Examples: Cash register, self-service cash register, pick-up terminal, digital shelf

5.4.3.1 Digital OTC shelf

Please refer to full WWKS2+ specification.

5.4.3.2 Digital self-service shelf

Please refer to full WWKS2+ specification.

5.4.3.3 Pick-up terminal

Please refer to full WWKS2+ specification.

5.4.3.4 Self-checkout cash register

A self-checkout cash register allows a customer to perform payment of items he collected or has been given. In case consultation or invalidation of serial numbers are required for specific items, these actions have to take place before the item is handed over to the patient.

A self-checkout cash register is usually operated by a pharmacy customer.

Relevant capabilities	Mandatory / Optional	Rowa Self-Checkout
Manage shopping carts (ShoppingCarts)	Mandatory	Supported
Scan Barcodes (BarcodeScan)	Mandatory	In development
Payment of shopping cards: Card terminal handling by device (Payment) or Card terminal handling by PIS (PISCardTerminal)	Mandatory	In development, supports deprecated solution
Shopping cart article master data (ShoppingCartArticleMaster)	Optional	In development, supports deprecated solution using ArticleInfoRequest/-Response, ArticlePriceRequest/-Response
Receipt layout by PIS (ReceiptLayout)	Optional	In development
Partial payment / payment with bonus points (PartialPayment)	Optional	In development

Typical role of the device in a WWKS 2+ connection: Client

```
Subscriber.Type in HelloRequest: „SelfCheckOut“
```

Basic workflow self-checkout

Purpose

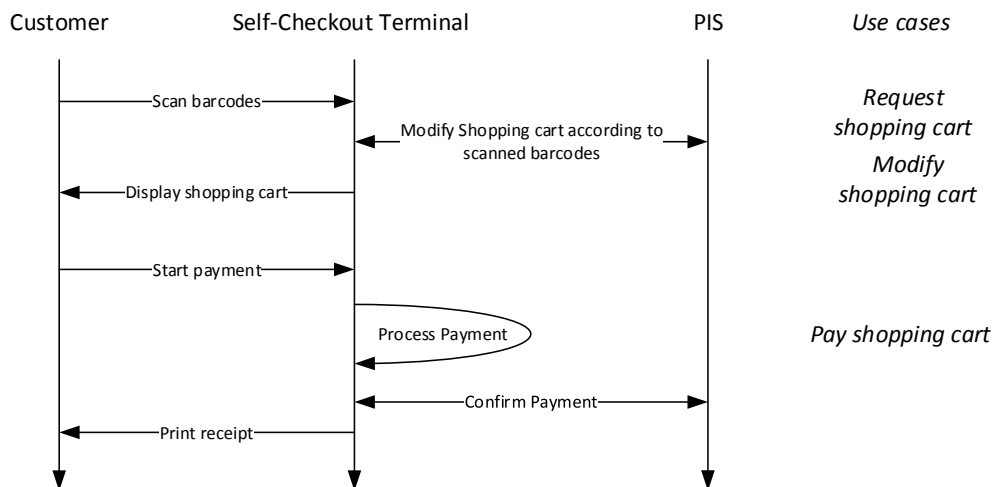
A customer within the pharmacy has picked several item from shelves or has been handed over items at a counter that he now wants to pay using a self-checkout terminal.

At the self-checkout terminal he scans all selected items and may present loyalty cards, coupons or other scanable items that may influence the final price of his shopping cart.

Preconditions

1. The PIS is able to map scanned barcodes to articles and other payment relevant items like loyalty cards or coupons.
2. The PIS is able to return prices for all items scanned into the shopping cart
3. The device or the PIS is able to process the actual payment
4. The device or the PIS is able to create a receipt layout
5. The device is able to print a receipt
6. Serial numbers for Rx products have been invalidated by the PIS when the items have been handed over to the customer

Standard sequence



Use cases (see chapter 7)

- Request shopping cart
- Modify shopping cart
- Pay shopping cart
- Shopping cart article master data

5.4.3.5 Information and Order terminals

Please refer to full WWKS2+ specification.

5.4.4 Screens

Please refer to full WWKS2+ specification.

5.4.5 Commercial excellence components

Please refer to full WWKS2+ specification.

6 Capabilities

A **capability** is a combination of supported WWKS 2+ messages, message attribute and use cases, that a device or system supports either fully or not at all.

Each capability has a unique name and a version number (cf. chapter 3.2.3 Version control).

6.1 Connection setup (Connect)

When setting up the connection (see chapter 4), the communication partners exchange information about the capabilities they support. *Connect* is the only use case that each WWKS 2+ system has to implement. It is not listed as capability in the *Hello* dialogue or as presupposed for other capabilities.

For connection setup, support of the following elements is required:

Name	<i>Connect</i>	Version	1.0
Presupposed capabilities		<i>none</i>	
Use Cases to support		Affected capabilities, if supported	
7.1 Connection establishment			
Messages to support		Elements/Attributes/Values to support	
HelloRequest / HelloResponse		Capability	
KeepAliveRequest / KeepAliveResponse		<i>None</i>	

6.2 Capabilities from WWKS2

To remain compatible to WWKS 2, WWKS 2+ supports the message based capabilities defined in WWKS 2.

Please refer to full WWKS2+ specification, not required for self-checkout terminals.

6.3 Stock management

Please refer to full WWKS2+ specification, not required for self-checkout terminals.

6.4 Item display

6.4.1 Display item information (ItemDisplay)

Use of this capability is deprecated for self-checkout terminals. Instead 6.5.2 Manage shopping cart (ShoppingCarts) and 6.5.8 Shopping cart article master data (ShoppingCartArticleMaster) capabilities should be used to retrieve item information.

Please refer to full WWKS2+ specification.

6.4.2 Price change indication (LivePriceUpdate)

Please refer to full WWKS2+ specification, not required for self-checkout terminals.

6.4.3 Relations between Items (ItemRelations)

Please refer to full WWKS2+ specification, not required for self-checkout terminals.

6.5 Sales transactions

6.5.1 Shopping cart update notification (ShoppingCartUpdateMessage)

Please refer to full WWKS2+ specification, not required for self-checkout terminals.

6.5.2 Manage shopping carts (ShoppingCarts)

Requests of shopping carts and add, modify or delete lines from them.

Name	<i>ShoppingCarts</i>	Version	1.0
Presupposed capabilities		Connect	
Use cases to support		Use cases affected, if supported	
7.9.1 Request shopping cart			
7.9.2 Modify shopping cart			
7.9.5 Terminate handling of shopping cart at a device			
Messages to support <i>Affected messages, if supported</i>		Elements/attributes/values to support	
ShoppingCartRequest / ShoppingCartResponse			
ShoppingCartUpdateRequest / ShoppingCartUpdateResponse		Add ChangeQuantity Close	
Affected data types		Elements/attributes/values to support	
Shopping Cart		ShoppingCart.Status, ShoppingCart.SalesPointId, ShoppingCartItem.ArticleId ShoppingCartItem.Name ShoppingCartItem.OrderedQuantity, ShoppingCartItem.Price ShoppingCartItem.Currency ShoppingCartItem.VAT	

6.5.3 Scan barcodes (BarcodeScan)

The device scans any kind of barcode and asks the PIS to modify the shopping cart according to the barcode type.

Name	<i>BarcodeScan</i>	Version	1.0
Presupposed capabilities		ShoppingCarts	
Use cases to support		Use cases affected, if supported	
7.9.3 Scan barcode for shopping cart			
Messages to support <i>Affected messages, if supported</i>		Elements/attributes/values to support	
ShoppingCartUpdateRequest/-Response		BarcodeScan	
Affected data types		Elements/attributes/values to support	
Shopping Cart		Barcode	

6.5.4 Delivery of shopping carts (ShoppingCartDelivery)

Please refer to full WWKS2+ specification, not required for self-checkout terminals.

6.5.5 Payment of shopping carts

Full payment of a shopping cart.

6.5.5.1 Card terminal handling by Device (Payment)

Card terminal device is handled by the device.

Name	<i>Payment</i>	Version	1.0
Presupposed capabilities		ShoppingCarts	
Use cases to support		Use cases affected, if supported	
7.9.6.1 Card terminal handling by device			
Messages to support <i>Affected messages, if supported</i>		Elements/attributes/values to support	
ShoppingCartUpdateRequest / ShoppingCartUpdateResponse		Payment	
Affected data types		Elements/attributes/values to support	
Shopping Cart		ShoppingCart.PaymentItem	

6.5.5.2 Card terminal handling by PIS (PISCardTerminal)

Though payment is handled by a device, the PIS is still in control of the card terminal used for the payment.

Name	<i>PISCardTerminal</i>	Version	1.0
Presupposed capabilities		Payment	
Use cases to support		Use cases affected, if supported	
7.9.6.2 Card terminal handling by PIS			
Messages to support <i>Affected messages, if supported</i>		Elements/attributes/values to support	
ShoppingCartUpdateRequest / ShoppingCartUpdateResponse		PaymentRequest	
Affected data types		Elements/attributes/values to support	
Shopping Cart		ShoppingCart.PaymentItem.TransactionDetails	

6.5.5.3 Receipt layout by PIS (ReceiptLayout)

The PIS provides receipt layout for payments handled by a device.

Name	<i>ReceiptLayout</i>	Version	1.0
Presupposed capabilities		Payment	
Use cases to support		Use cases affected, if supported	
7.9.6.3 Receipt layout by PIS			
Messages to support <i>Affected messages, if supported</i>		Elements/attributes/values to support	
ShoppingCartUpdateRequest / ShoppingCartUpdateResponse			
Affected data types		Elements/attributes/values to support	
Shopping Cart		ShoppingCart.Payment.ReceiptWidth <i>and/or</i> ShoppingCart.PaymentRequest.ReceiptWidth ShoppingCart.PaymentItem.Receipt	

6.5.5.4 Partial payment / payment with bonus points (PartialPayment)

The PIS supports partial payment and partial or full payment with alternate currencies like bonus points.

Name	<i>PartialPayment</i>	Version	1.0
Presupposed capabilities		Payment	
Use cases to support		Use cases affected, if supported	
7.9.6.4 Partial payment / payment with bonus points			
Messages to support <i>Affected messages, if supported</i>		Elements/attributes/values to support	
ShoppingCartUpdateRequest / ShoppingCartUpdateResponse			
Affected data types		Elements/attributes/values to support	
Shopping Cart		ShoppingCart.PaymentItem.Currency	

6.5.6 Transfer of shopping carts (ShoppingCartTransfer)

Request of a device to transfer a shopping cart to another sales point

Name	<i>ShoppingCartTransfer</i>	Version	1.0
Presupposed capabilities		ShoppingCarts	
Use cases to support		Use cases affected, if supported	
7.9.8 Transfer shopping cart to another sales point			
Messages to support <i>Affected messages, if supported</i>		Elements/attributes/values to support	
ShoppingCartRequest / ShoppingCartResponse			
ShoppingCartUpdateRequest / ShoppingCartUpdateResponse		Action={Transfer}	
Affected data types		Elements/attributes/values to support	
Shopping Cart		ShoppingCart.SalesPointId	

6.5.7 Item selection (ArticleSelection)

Please refer to full WWKS2+ specification, not required for self-checkout terminals.

6.5.8 ShoppingCart article master data (ShoppingCartArticleMaster)

The PIS includes selected article master data for items in a shopping cart so that the device does not has to query them separately.

Name	<i>ShoppingCartArticleMaster</i>	Version	1.0
Presupposed capabilities		ShoppingCarts	
Use cases to support		Use cases affected, if supported	
7.9.12 Shopping cart with item master data		7.9.1 Request shopping cart 7.9.2 Modify shopping cart	
Messages to support <i>Affected messages, if supported</i>		Elements/attributes/values to support	
<i>ShoppingCartResponse</i> <i>ShoppingCartUpdateResponse</i>		ShoppingCart.ShoppingCartItem.Article ShoppingCart.ShoppingCartItem.Article	
Affected data types		Elements/attributes/values to support	
ShoppingCart.ShoppingCartItem.Article ShoppingCart.ShoppingCartItem.Article. PriceInformation ShoppingCart.ShoppingCartItem.Article.Tag		Name, DosageForm, PackagingUnit	

7 Use cases

Pharmacy IT system ownership of workflows

The pharmacy IT system owns and manages all workflows in the pharmacy. The communication between the devices in the pharmacy should always be coordinated by the pharmacy IT system to ensure consistent workflow logic and behavior.

Direct communication between different devices makes sense as a fallback solution if the PIS does not support the coordinating function for the respective use case.

Capabilities

When establishing the WWKS 2+ connection, both sides exchange information which of the use case described in this document they support. The use cases are bundled to capabilities. Each system has to support all elements of a capability (see chapter 6 Capabilities).

7.1 Connection establishment

Purpose

This use case is used to ensure that a connected socket is really used to talk WWKS 2+ and to exchange information about the capabilities supported by both sides.

Only the capabilities that are supported by the current configuration are communicated. Basically supported abilities, which still have to be "switched on", are not listed.

In difference to WWKS 2, the listing of supported skills is mandatory in WWKS 2+.

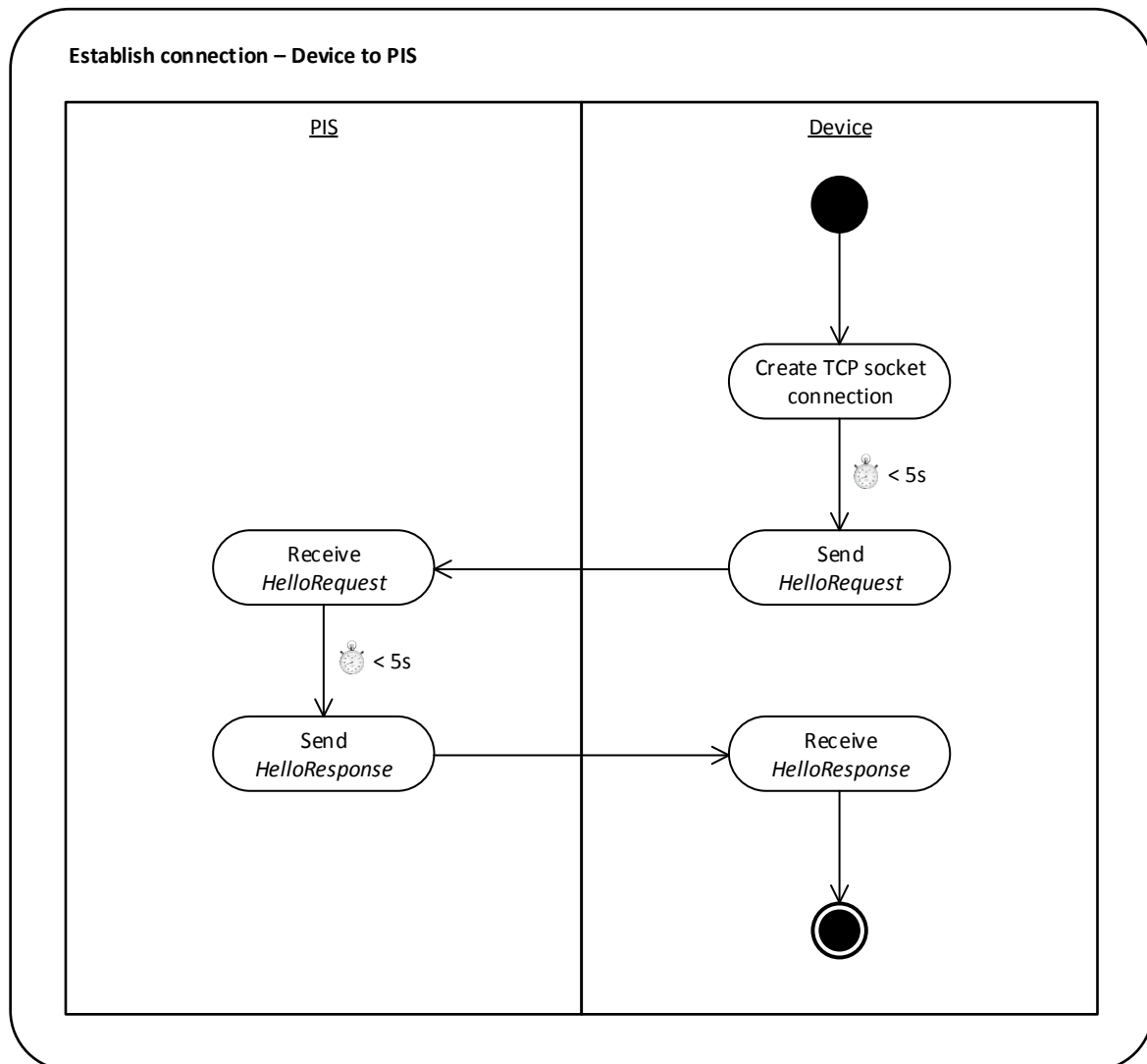
If a side requires certain capabilities of the other side, but these are not listed by it, it will terminate the socket connection after completion of the Hello dialog. The missing capability is not communicated via the interface, but via a feedback message to the user of the respective system.

Both sides do not accept any other WWKS 2+ messages before the connection has been successfully established.

Connection establishment by the PIS

Please refer to full WWKS 2+ specification. For self-checkout terminals, usually the device establishes the connection.

Connection establishment by the device



```

<WWKS Version="2.0" TimeStamp="2013-04-16T11:14:00Z">
  <HelloRequest Id="1">
    <Subscriber Id="431" Type="SelfCheckout" Manufacturer="DeviceVendor"
      ProductInfo="MegaDevice" VersionInfo="1.4.0">
      <Capability Name="ShoppingCarts"/>
      <Capability Name="Payment"/>
      <Capability Name="BarcodeScan"/>
      <Capability Name="ShoppingCartArticleMaster"/>
    </Subscriber>
  </HelloRequest>
</WWKS>
  
```

```
<WWKS Version="2.0" TimeStamp="2013-04-16T11:14:00Z">
  <HelloResponse Id="1">
    <Subscriber Id="100" Type="WMS" Manufacturer="WMSVendor"
      ProductInfo="SuperPIS" VersionInfo="1.3">
      <Capability Name="ShoppingCarts"/>
      <Capability Name="Payment"/>
      <Capability Name="BarcodeScan"/>
      <Capability Name="ShoppingCartArticleMaster"/>
      <Capability Name="ReceiptLayout"/>
    </Subscriber>
  </HelloResponse>
</WWKS>
```

Multiple devices sharing a socket connection

It is generally allowed to connect several devices of a homogeneous group to the PIS via the same socket connection.

In such a case, the `HelloResponse` (or `HelloRequest` if the connection is established by the device) contains several subscriber elements, each of which describes a connected device.

Device groups that have to be configured differently on the PIS side or are integrated in workflows in different ways should use separate connections to the PIS.

```
<WWKS Version="2.0" TimeStamp="2013-04-16T11:14:00Z">
  <HelloRequest Id="1">
    <Subscriber Id="431" Type="SelfCheckOut" Manufacturer="TerminalVendor"
      ProductInfo="MegaCheckOut" VersionInfo="1.5.0">
      <Capability Name="ShoppingCarts"/>
      <Capability Name="Payment"/>
      <Capability Name="BarcodeScan"/>
      <Capability Name="ReceiptLayout"/>
      <Capability Name="ShoppingCartArticleMaster"/>
    </Subscriber>
    <Subscriber Id="432" Type="SelfCheckOut" Manufacturer="TerminalVendor"
      ProductInfo="MegaCheckOut" VersionInfo="1.3.0">
      <Capability Name="ShoppingCarts"/>
      <Capability Name="Payment"/>
      <Capability Name="BarcodeScan"/>
      <Capability Name="ShoppingCartArticleMaster"/>
    </Subscriber>
  </HelloRequest>
</WWKS>
```

```
<WWKS Version="2.0" TimeStamp="2013-04-16T11:14:00Z">
  <HelloResponse Id="1">
    <Subscriber Id="100" Type="WMS" Manufacturer="WMSVendor"
ProductInfo="SuperPIS" VersionInfo="1.3">
      <Capability Name="ShoppingCarts"/>
      <Capability Name="ShoppingCartDelivery"/>
      <Capability Name="Payment"/>
      <Capability Name="ShoppingCartArticleMaster"/>
    </Subscriber>
  </HelloResponse>
</WWKS>
```

Redundant PIS connections

In case of a failure of a PIS server, the PIS may switch to an alternative server and establish a new connection to the device if it acts as client in the WWKS 2+ connection.

If the PIS acts as server in the WWKS 2+ connection, the WWKS 2+ protocol does not define any failure procedure.

Insufficient capabilities

If the list of capabilities transmitted in `HelloRequest` or `HelloResponse` is not sufficient for the respective opposing side, it terminates the socket connection.

There is no notification of the reason for the connection termination via WWKS 2+. Instead, the respective system informs the user about the missing capabilities via a user message and a log entry. The user contacts the manufacturer of the other side and informs it of the error message.

Adaptation of abilities in the HelloResponse

The server may adapt the list of capabilities it transmits in the `HelloResponse` to the capabilities sent by the client in the `HelloRequest`, if this is helpful for clarifying the use cases used.

Device number already in use

If the device number specified by the device in the `Subscriber.ID` is already in use, the PIS terminates the socket connection.

There is no notification of the reason for the connection termination via WWKS 2+. Instead, the PIS informs the user about a user message and a log entry about the colliding device number. The user contacts the manufacturer of the device and informs them about the error message.

Maximum reaction times

Processing step	Maximum Duration	Reaction of other side if no message is sent in time
Connection establishment until sending HelloRequest	5s	Terminates socket connection
Receipt of HelloRequest until sending HelloResponse	5s	Terminates socket connection

7.2 System state

Please refer to full WWKS 2+ specification, currently not specified for self-checkout terminals.

7.3 Item master data

Please refer to full WWKS 2+ specification, not required for self-checkout terminals.

This interface section defines messages for transferring article master data. The use of these messages is deprecated for self-checkout terminals. Please refer to use case 7.9.12 Shopping cart with article master data instead.

7.4 Prices

WWKS 2+ assumes that the PIS manages multiple price categories that define the item price currently valid in different environments.

The actual price determination within the PIS is not relevant for the WWKS 2+ interface. Only prices that are displayed or used in a specific context are exchanged via the interface.

Prices that are valid just for specific customers or customer groups are usually handled in the context of a shopping cart and are not in scope of the use cases in this chapter.

WWKS 2+ predefines just a few price categories that have a specific meaning in the context of the interface. The PIS may define and transmit additional categories.

Price category	Meaning
RRP	Recommended Retail Price or list price as defined by the manufacturer.
Standard	Default price in the pharmacy usually displayed on a price label visible to customers in the pharmacy.
Offer	Current offer price that differs from the standard price.
Loyalty	Special price for a customer group.

Prices for shopping cart lines are managed within the shopping cart.

This interface section defines messages for generic price updates which are deprecated for self-checkout terminals. Article price information are transmitted via use case 7.9.12 Shopping cart with article master data instead.

For use cases in this section, please refer to full WWKS 2+ specification

7.5 Marketing data

Please refer to full WWKS 2+ specification, currently not supported for self-checkout terminals.

7.6 Storage

Please refer to full WWKS 2+ specification, not required for self-checkout terminals.

7.7 Serial number handling (FMD)

Please refer to full WWKS 2+ specification, not required for self-checkout terminals.

7.8 Check good receipts

Please refer to full WWKS 2+ specification, not required for self-checkout terminals.

7.9 Sales transactions

The PIS uses sales transactions to manage the current state of a sale and delivery of a shopping cart of one or multiple items to a patient.

The PIS is the owner of the sales transaction at all times. All changes to a sales transaction by a device must be authorized or can be rejected by the PIS. The device must be prepared to deal with any rejection of a sales transaction change.

Business Records

For each relevant modification of the sales transaction, the PIS creates a business recording according to accounting standards. Creation of a business record may lead to cancelation and creation of a new sales transaction. The PIS assures that the identification number of the sales transaction used in this interface does not change over time. This means that a business record number usually cannot be used as sales transaction identification number.

Devices that fully handle sales transactions internally are out of scope of this interface specification.

Point of sales assigned to a sales transaction

A sales transaction is always assigned to either no or exactly one point of sale (SalesPoint). At a point of sale, just one patient is served at a time. Multiple devices may be available at a point of sale that work on the same sales transaction in parallel.

Typical sales points in the pharmacy are:

- A PIS cash register in combination with one or multiple digital shelves,
- A self-service terminal or cash register,
- A pick-up terminal.

If a patient is served further at another point of sale, the PIS has to explicitly transfer the sales transaction to this point.

A device may be designed to support multiple points of sale, such as a digital display located between two cash registers. In this case, the device must ensure that, for each action that can change a sales transaction, the operator selects for which point of sale, and therefore for which sales transaction, the change is made. The device is therefore always assigned to a maximum of one sales location for a particular point in time, but can switch back and forth between several locations.

This interface uses **shopping carts** to model the data of a sales transaction.

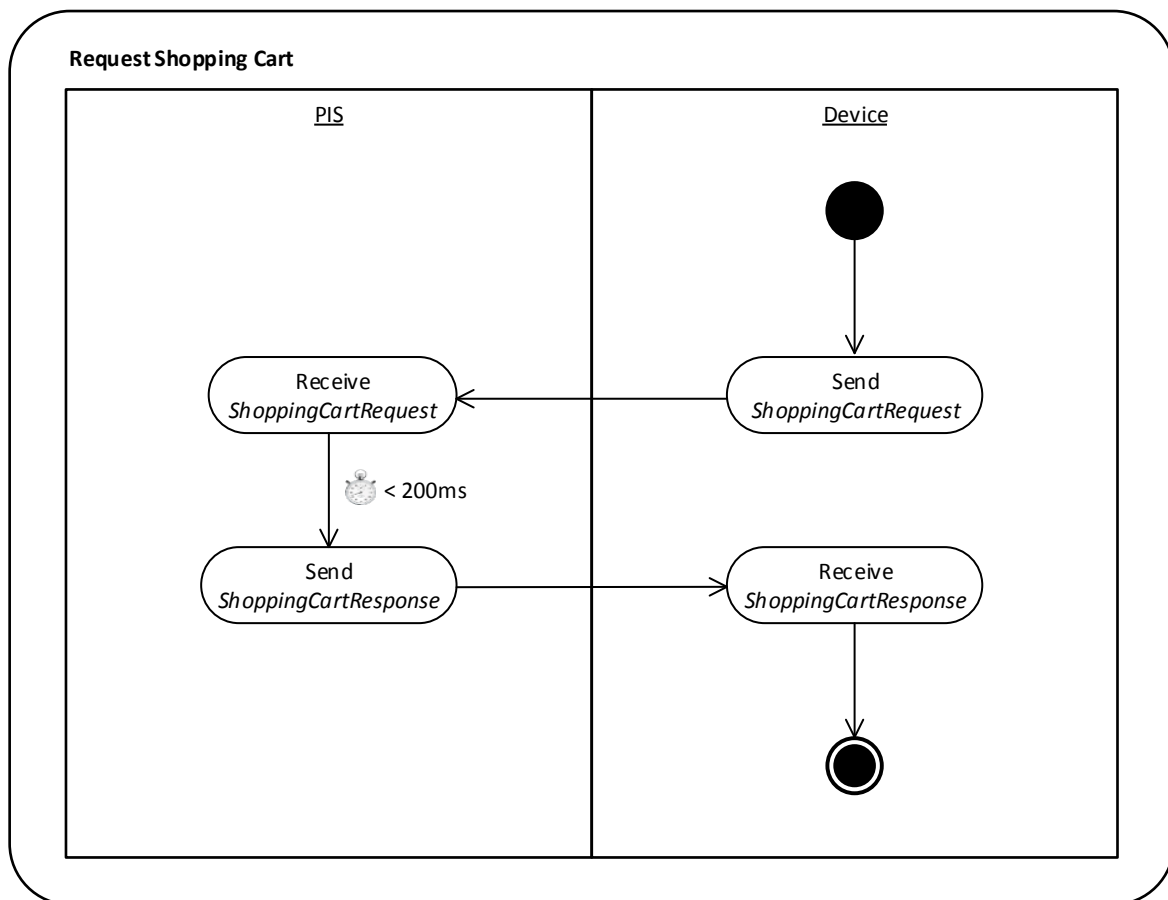
7.9.1 Request shopping cart

Purpose

A device can request a shopping cart from the PIS. This can either be empty (for a new sales transaction) or contain information about an existing transaction.

The type of the sales transaction requested depends on the search criteria transmitted for the shopping cart.

Default procedure



Request a new shopping cart

A device may ask for a new shopping cart by defining no search criteria in the request.

In case the PIS knows to which sales point the device belongs to, it may already assign the shopping cart to a sales point by setting the *SalesPointId* attribute in the *ShoppingCartResponse*. The device may as well request assignment to its sales point by sending a *ShoppingCartUpdateRequest* for the sales transaction.

```
<WWKS Version="2.0" TimeStamp="2013-04-16T11:14:00Z">
  <ShoppingCartRequest Id="1103" Source="701" Destination="100">
  </ShoppingCartRequest>
</WWKS>
```

```
<WWKS Version="2.0" TimeStamp="2013-04-16T11:14:00Z">
  <ShoppingCartResponse Id="1103" Source="100" Destination="701">
    <ShoppingCart Id="SC-2345" SalesPointId="3" Status="Active"/>
  </ShoppingCartResponse>
</WWKS>
```

Request current shopping cart of a point of sale

Please refer to full WWKS 2+ specification, not required for self-checkout terminals.

Request transfer shopping cart to point of sale

Please refer to full WWKS 2+ specification, not required for self-checkout terminals.

Query shopping carts using a pick-up code

Please refer to full WWKS 2+ specification, not required for self-checkout terminals.

Display language

In case the device supports multiple display languages, it may transmit the current language used by the customer in the request. The PIS may return line item names and descriptions in this language, if supported. In case the PIS does not support multiple languages or the requested language, it will return names and descriptions in its default language.

```
<WWKS Version="2.0" TimeStamp="2013-04-16T11:14:00Z">
  <ShoppingCartRequest Id="1103" Source="701" Destination="100"
    DisplayLanguage="fr">
  </ShoppingCartRequest>
</WWKS>
```

Query shopping carts using a pick-up code

Please refer to full WWKS 2+ specification, not required for self-checkout terminals.

Query shopping cart using customer identification attributes

Please refer to full WWKS 2+ specification, not required for self-checkout terminals.

Denial of request by the PIS

The PIS may reject the request by sending an empty response to the device. It may use the `Description` attribute to give a human readable explanation that the device may display to the patient.

```
<WWKS Version="2.0" TimeStamp="2013-04-16T11:14:00Z">
  <ShoppingCartRequest Id="1103" Source="701" Destination="100">
    <Criteria ShoppingCartId="48779"/>
  </ShoppingCartRequest>
</WWKS>
```

```
<WWKS Version="2.0" TimeStamp="2013-04-16T11:14:00Z">
  <ShoppingCartResponse Id="1103" Source="100" Destination="701"
    Description="Please talk with a pharmacist."/>
</WWKS>
```

Maximum reaction times

Processing step	Maximum Duration	Reaction of other side if no message is sent in time
Receipt of ShoppingCartRequest/ ShoppingCartUpdateRequest until sending Response	200ms	Assumption that request wasn't granted.

7.9.2 Modify shopping cart

Purpose

A device asks the PIS to modify an existing shopping cart according to its request.

Default procedure

There different possible types of modifications. The device specifies the type of the modification by adding an update request element to the `ShoppingCartUpdateRequest`.

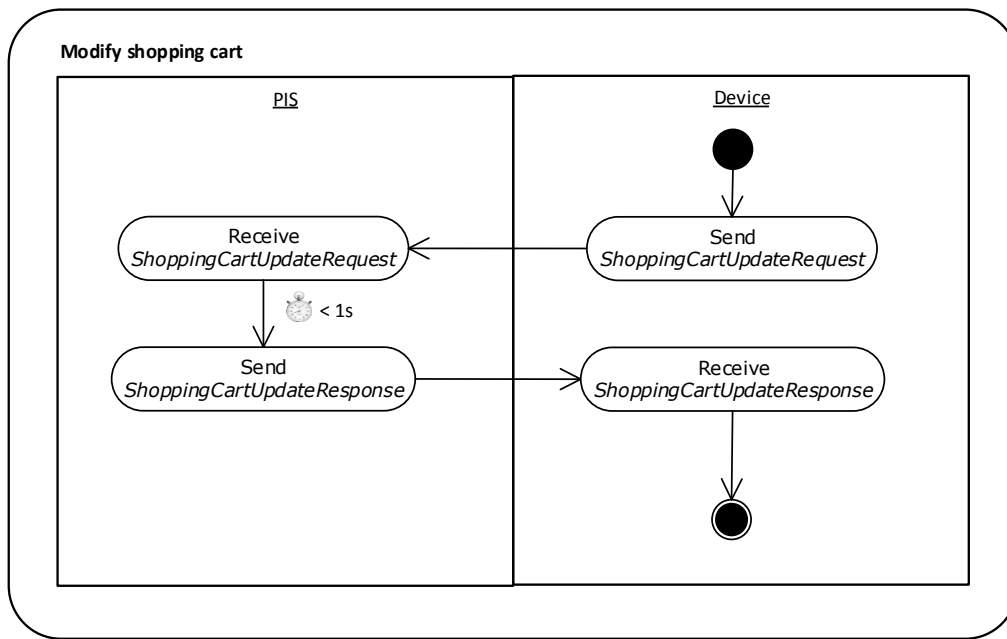
In case the IT system accepts the request, it returns the modified shopping cart.

To simplify error analysis, the device may provide its view of the updated shopping cart in the request as well. The IT system usually ignores the shopping cart provided by the device and updates the cart based on its local state.

Update element	Shopping Cart Modification intended by the device	Shopping Cart elements that might be affected	Used by Rowa device
BarcodeScanned	Barcode scanned (see use case 7.9.3 Scan barcode for shopping card)	ShoppingCartItem elements added or OrderedQuantity changed	Self-Checkout
ChangeQuantity	Remove or modify lines	ShoppingCartItem elements removed or OrderedQuantity changed	Self-Checkout
Payment	Customer has paid the shopping cart fully or partially	ShoppingCartItem.PaidQuantity (<i>deprectated</i>) ShoppingCart.PaymentItem	Self-Checkout
Payment Request	Request to the PIS to handle payment of the shopping cart	ShoppingCart.PaymentItem	Self-Checkout
Close	Finish handling of shopping cart at a device.		Self-Checkout,

WWKS2 defines further update elements Add, Delivery, Transfer and CartUpdate which are not used by a self-checkout terminal. Please refer to full WWKS 2+ specification.

In its answer, the PIS tells the device if it has accepted the modification request and returns the updated shopping cart. The shopping cart may contain as well other modifications, e.g. updated prices. The device has to take over fully the shopping cart returned by the PIS even if this differs from the previous state plus the modifications requested.



Adding a line to the shopping cart

Please refer to full WWKS 2+ specification, not used by self-checkout terminals.

Changing the ordered quantity of a shopping cart line

The device sends a `ShoppingCartUpdateRequest` with update element `ChangeQuantity` and the shopping cart with the modified `OrderedQuantity` and the previous amount in `LastOrderedQuantity`.

```

<WWKS Version="2.0" TimeStamp="2013-04-16T11:14:00Z">
  <ShoppingCartUpdateRequest Id="1107" Source="701"
    Destination="100" ShoppingCartId="SC-2345">
    <ChangeQuantity ArticleId="1234" OrderedQuantity="3"
      LastOrderedQuantity="2"/>
  </ShoppingCartUpdateRequest>
</WWKS>
  
```

```

<WWKS Version="2.0" TimeStamp="2013-04-16T11:14:00Z">
  <ShoppingCartUpdateResponse Id="1107" Source="100"
    Destination="701">
    <ShoppingCart Id="SC-2345" Status="Active" SalesPointId="701">
      <ShoppingCartItem ArticleId="1234" OrderedQuantity="3" .../>
      <ShoppingCartItem ArticleId="5678" OrderedQuantity="2" .../>
    </ShoppingCart>
    <UpdateResult Status="Updated"/>
  </ShoppingCartUpdateResponse>
</WWKS>
  
```

Remove a line from the shopping cart

The device sends a `ShoppingCartUpdateRequest` with update element `ChangeQuantity` and in which the order quantity of the line item it likes to delete is set to zero.

If the PIS accepts the request, it answers with a shopping cart that no longer contains the line item.

```
<WWKS Version="2.0" TimeStamp="2013-04-16T11:14:00Z">
  <ShoppingCartUpdateRequest Id="1108" ShoppingCartId="Sc-" Source="701"
    Destination="100" ShoppingCartId="SC-2345">
    <ChangeQuantity ArticleId="1234" OrderedQuantity="0"
      LastOrderedQuantity="1"/>
  </ShoppingCartUpdateRequest>
</WWKS>
```

```
<WWKS Version="2.0" TimeStamp="2013-04-16T11:14:00Z">
  <ShoppingCartUpdateResponse Id="1108" Source="100"
    Destination="701">
    <ShoppingCart Id="SC-2345" Status="Active" SalesPointId="701">
      <ShoppingCartItem ArticleId="5678" OrderedQuantity="2" .../>
    </ShoppingCart>
    <UpdateResult Status="Updated"/>
  </ShoppingCartUpdateResponse>
</WWKS>
```

Partial acceptance of a modification

The WWKS 2+ interface does not support partial acceptance of a modification. In case the PIS is not able to accept all modifications in the request it has to deny the modification as whole.

Denial of the modification

The PIS may deny the modification of the shopping cart requested by the device. In this case it sends a `ShoppingCartUpdateResponse` with `UpdateResult.Status="NotUpdated"` and may give a human readable explanation in `UpdateResult.Description` that the device may show to the customer.

The response contains the current state of the shopping cart without the modification requested. The current state of the shopping cart may be different to what the device assumes to be the last state before the modification and the device has to update its copy of the shopping cart.

In case of a denial the PIS does not send a `ShoppingCartUpdateMessage` to inform other devices.

```
<WWKS Version="2.0" Timestamp="2013-04-16T11:14:00Z">
  <ShoppingCartUpdateRequest Id="1110" Source="701"
    Destination="100" ShoppingCartId="SC-2345">
    <ChangeQuantity ArticleId="5678" OrderedQuantity="2"
      LastOrderedQuantity="2"/>
  </ShoppingCartUpdateRequest>
</WWKS>
```

```
<WWKS Version="2.0" Timestamp="2013-04-16T11:14:00Z">
  <ShoppingCartUpdateResponse Id="1110" Source="100"
    Destination="701">
    <ShoppingCart Id="SC-2345" Status="Active" SalesPointId="701">
      <ShoppingCartItem ArticleId="5678" OrderedQuantity="2" .../>
    </ShoppingCart>
    <UpdateResult Status="NotUpdated" Description="Item not known"/>
  </ShoppingCartUpdateResponse>
</WWKS>
```

Possible reasons for a denial of the modification request by the PIS are:

- Shopping cart id invalid or unknown
- Shopping cart no longer assigned to the sales point (timeout)
- Article id unknown
- Article not available for this device
- No stock
- Not allowed to change ordered quantity, line item is already delivered or paid
- ...

Change of display language

The device informs the PIS that the user has changed the display language of the device and requests to transmit line item names and descriptions in the new language by sending a `ShoppingCartUpdateRequest` with update request element `Language`.

If accepted, the PIS returns the updated shopping cart and changes the `DisplayLanguage` attribute of the shopping cart. If rejected, the PIS returns the unmodified shopping cart.

```
<WWKS Version="2.0" Timestamp="2013-04-16T11:14:00Z">
  <ShoppingCartUpdateRequest Id="1110" Source="701"
    Destination="100" ShoppingCartId="SC-2345">
    <Language DisplayLanguage="en"/>
  </ShoppingCartUpdateRequest>
</WWKS>
```



```
<WWKS Version="2.0" TimeStamp="2013-04-16T11:14:00Z">
  <ShoppingCartUpdateResponse Id="1110" Source="100"
    Destination="701" Action="Order">
    <ShoppingCart Id="SC-2345" Status="Active" SalesPointId="701"
      DisplayLanguage="en">
      <ShoppingCartItem ArticleId="1234" Name="Drug 1234" .../>
    </ShoppingCart>
    <UpdateResult Status="Updated"/>
  </ShoppingCartUpdateResponse>
</WWKS>
```

Device provides its idea of modify shopping cart for error analysis

Optionally, to simplify error analysis, the device may provide its idea of how the result of the shopping cart update should look like by providing a ShoppingCart element within the ShoppingCartUpdateRequest. The IT system will usually ignore this element and update the shopping cart based on its internal state and the update element of the request.

```
<WWKS Version="2.0" TimeStamp="2013-04-16T11:14:00Z">
  <ShoppingCartUpdateRequest Id="1107" Source="701"
    Destination="100" ShoppingCartId="SC-2345">
    <ChangeQuantity ArticleId="1234" OrderedQuantity="3"
      LastOrderedQuantity="2"/>
    <ShoppingCart Id="SC-2345" Status="Active" SalesPointId="701">
      <ShoppingCartItem ArticleId="1234" OrderedQuantity="3" .../>
      <ShoppingCartItem ArticleId="5678" OrderedQuantity="2" .../>
    </ShoppingCart>
  </ShoppingCartUpdateRequest>
</WWKS>
```

Shopping cart update notification

In case the PIS supports use case 7.9.9 Inform about shopping cart modifications the device asking for a shopping cart modification receives as well a ShoppingCartUpdateMessage.

Maximum reaction times

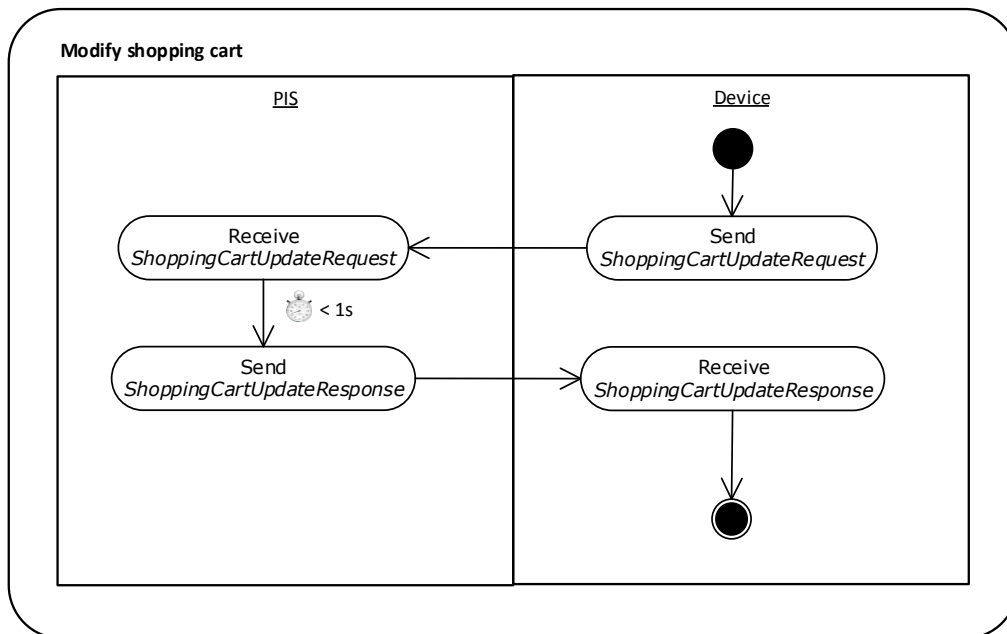
Processing step	Maximum Duration	Reaction of other side if no message is sent in time
Receiving ShoppingCartUpdateRequest until sending ShoppingCartUpdateResponse	1s	Assume that modification has been denied.

7.9.3 Scan barcode for shopping cart

Purpose

The device transfers to the WWS a scanned barcode that a user of the device has scanned with the intention to modify a shopping cart already assigned to the device.

Default procedure



The message flow is the same as for the Modify shopping cart use case, with the device sending a `ShoppingCartUpdateRequest` and the PIS answering with a `ShoppingCartUpdateResponse`. The device just uses specific values for the `Action` attribute to indicate a barcode scan and transmits the barcode content in the `Barcode` attribute.

Depending on the workflow at the device, the device may or may not know the type of the barcode scanned. If the PIS is able to differentiate different types of codes from the barcode content, the device may support scanning of any kind barcode at any time.

In case the PIS requires the device to transfer the type of the barcode, the device has to guide the user to scan a specific type of code at dedicated points within the workflow, so that is able to transfer the barcode type as well. The device then uses the actions defined below to indicate the type of the code to the PIS. Default scan action is to scan a pack, therefore the interface does not differentiate between unknown barcode type and article barcode scanned. Both are handled with the `BarcodeScanned` action.

The WWKS 2+ protocol does not provide the possibility to exchange information if and when the PIS requires information about the barcode type.

Types of scanned barcodes supported by WWKS 2+:

Barcode type	BarcodeScanned.Type	PIS Action
Unknown barcode type	Type="Unknown"	Determine barcode type and act according to types listed below
Article barcode	Type="Article"	Identify article id matching the barcode. If already in shopping cart, increase <code>OrderedQuantity</code> by 1, otherwise add shopping cart item for this article id and <code>OrderedQuantity</code> 1.
Loyalty card	Type="LoyaltyCard"	Set <code>CustomerId</code> attribute. Modify prices if customer has special pricing conditions.
Discount coupon	Type="Coupon"	Add shopping cart item for the discount. Either apply negative price to this line item or modify price of affected article line item.
Employee badge	<i>To be defined</i>	Set <code>SalesPersonId</code> attribute
Id card	<i>To be defined</i>	Set <code>CustomerId</code> attribute. Modify prices if customer has special pricing conditions.

The PIS answers with an updated shopping cart that is modified according to the barcode scanned.

Article barcode scanned that does not yet exist in the shopping cart

The device uses the `BarcodeScanned` update element with `type="Article"` or `type="Unknown"`. The PIS adds a line item to the shopping cart.

In the example, the barcode "-07464882" belongs to article id "3456".

<pre><WWKS Version="2.0" TimeStamp="2013-04-16T11:14:00Z"> <ShoppingCartUpdateRequest Id="1104" Source="701" Destination="100" ShoppingCartId="SC-2345"> <BarcodeScanned Type="Article" Barcode="-07464882"> </BarcodeScanned> </ShoppingCartUpdateRequest> </WWKS></pre>	
	<pre><WWKS Version="2.0" TimeStamp="2013-04-16T11:14:00Z"> <ShoppingCartUpdateResponse Id="1104" Source="100" Destination="701"> </ShoppingCartUpdateResponse> </WWKS></pre>

```

<ShoppingCart Id="SC-2345" Status="Active" SalesPointId="701"
    Currency="EUR">
  <ShoppingCartItem ArticleId="1234" Name="Article-1234"
    OrderedQuantity="1" DispensedQuantity="0" PaidQuantity="0"
    Price="9.90" VAT=19.0"/>
  <ShoppingCartItem ArticleId="5687" Name="Article-5687"
    OrderedQuantity="2" DispensedQuantity="0" PaidQuantity="0"
    Price="8.90" VAT=19.0"/>
  <ShoppingCartItem ArticleId="3456" Name="Article-3456"
    OrderedQuantity="1" DispensedQuantity="0" PaidQuantity="0"
    Price="17.55" VAT=19.0"/>
</ShoppingCart>
<UpdateResult Status="Updated" />
</ShoppingCartUpdateResponse>
</WWKS>

```

Article barcode scanned that does not yet exist in the shopping cart

The device uses the `BarcodeScanned` update element. The PIS increases the `OrderedQuantity` of the article line item by 1.

In the example, the barcode "5000347087141" belongs to article id "1234".

```

<WWKS Version="2.0" TimeStamp="2013-04-16T11:14:00Z">
  <ShoppingCartUpdateRequest Id="1104" Source="701"
    Destination="100" ShoppingCartId="SC-2345">
    <BarcodeScanned Barcode="5000347087141" Type="Unknown">
  </ShoppingCartUpdateRequest>
</WWKS>

```

```

<WWKS Version="2.0" TimeStamp="2013-04-16T11:14:00Z">
  <ShoppingCartUpdateResponse Id="1104" Source="100"
    Destination="701">
    <ShoppingCart Id="SC-2345" Status="Active" SalesPointId="701"
      Currency="EUR">
      <ShoppingCartItem ArticleId="1234" Name="Article-1234"
        OrderedQuantity="2" DispensedQuantity="0" PaidQuantity="0"
        Price="9.90" VAT=19.0"/>
      <ShoppingCartItem ArticleId="5687" Name="Article-5687"
        OrderedQuantity="2" DispensedQuantity="0" PaidQuantity="0"
        Price="8.90" VAT=19.0"/>
    </ShoppingCart>
  </ShoppingCartUpdateResponse>
</WWKS>

```

```

    <UpdateResult Status="Updated" />
  </ShoppingCartUpdateResponse>
</WWKS>

```

Scan of a loyalty card

In case the PIS recognizes the barcode as a loyalty card or the device has defined type="LoyaltyCard" in the update element, the PIS set the ID of the related customer in the shopping cart, modifies the line item prices in case the customer has special price conditions and may add additional data to the shopping cart the device shall print on the receipt (e.g. current balance of bonus points account).

```

<WWKS Version="2.0" TimeStamp="2013-04-16T11:14:00Z">
  <ShoppingCartUpdateRequest Id="1104" Source="701"
    Destination="100" ShoppingCartId="SC-2345">
    <BarcodeScanned Type="LoyaltyCard" Barcode="LC74583">
  </ShoppingCartUpdateRequest>
</WWKS>

```

```

<WWKS Version="2.0" TimeStamp="2013-04-16T11:14:00Z">
  <ShoppingCartUpdateResponse Id="1104" Source="100"
    Destination="701">
    <ShoppingCart Id="SC-2345" Status="Active" SalesPointId="701"
      Currency="EUR" CustomerId="74583">
      <ShoppingCartItem ArticleId="1234" Name="Article-1234"
        OrderedQuantity="1" DispensedQuantity="0" PaidQuantity="0"
        Price="9.90" VAT=19.0"/>
      <ShoppingCartItem ArticleId="5687" Name="Article-5687"
        OrderedQuantity="2" DispensedQuantity="0" PaidQuantity="0"
        Price="8.90" VAT=19.0"/>
      <ShoppingCartInfo Item="Welcome" Value="Hello Ms. Smith"/>
      <ShoppingCartInfo Item="BonusPointAccount" Value="425 Points"/>
    </ShoppingCart>
    <UpdateResult Status="Updated" />
  </ShoppingCartUpdateResponse>
</WWKS>

```

Scan of a coupon

In case the PIS recognizes the barcode as a discount coupon or the device has defined type="Coupon" in the update element, the PIS adds an additional line item to the shopping cart for the coupon. It either applies a negative price to the coupon line item or reduces the prices of the related article line items according to the discount.

```
<WWKS Version="2.0" TimeStamp="2013-04-16T11:14:00Z">
  <ShoppingCartUpdateRequest Id="1104" Source="701" Destination="100"
    ShoppingCartId="SC-2345">
    <BarcodeScanned Type="Coupon" Barcode="CP99457742">
  </ShoppingCartUpdateRequest>
</WWKS>
```

```
<WWKS Version="2.0" TimeStamp="2013-04-16T11:14:00Z">
  <ShoppingCartUpdateResponse Id="1104" Source="100" Destination="701">
    <ShoppingCart Id="SC-2345" Status="Active" SalesPointId="701"
      Currency="EUR">
      <ShoppingCartItem ArticleId="1234" Name="Article-1234"
        OrderedQuantity="1" DispensedQuantity="0" PaidQuantity="0"
        Price="9.90" VAT=19.0"/>
      <ShoppingCartItem ArticleId="CP1234-10" Name="1234 10% discount"
        OrderedQuantity="1" DispensedQuantity="0" PaidQuantity="0"
        Price="-0.99" VAT=19.0"/>
      <ShoppingCartItem ArticleId="5687" Name="Article-5687"
        OrderedQuantity="2" DispensedQuantity="0" PaidQuantity="0"
        Price="8.90" VAT=19.0"/>
    </ShoppingCart>
    <UpdateResult Status="Updated" />
  </ShoppingCartUpdateResponse>
</WWKS>
```

Unknown Barcode

In case the PIS does not know the barcode scanned or is not able to process it, it rejects the update and returns the current state of the shopping cart in the ShoppingCartUpdateResponse.

It may give a human readable explanation in UpdateResult.Description that the device may show to the customer.

```
<WWKS Version="2.0" TimeStamp="2013-04-16T11:14:00Z">
  <ShoppingCartUpdateRequest Id="1104" Source="701"
    Destination="100" ShoppingCartId="SC-2345">
    <BarcodeScanned Barcode="8934587345">
  </ShoppingCartUpdateRequest>
</WWKS>
```

<pre> </ShoppingCartUpdateRequest> </WWKS> </pre>	
<pre> <WWKS Version="2.0" TimeStamp="2013-04-16T11:14:00Z"> <ShoppingCartUpdateResponse Id="1104" Source="100" Destination="701"> <ShoppingCart Id="SC-2345" Status="Active" SalesPointId="701" Currency="EUR"> <ShoppingCartItem ArticleId="1234" Name="Article-1234" OrderedQuantity="1" DispensedQuantity="0" PaidQuantity="0" Price="9.90" VAT=19.0"/> <ShoppingCartItem ArticleId="5687" Name="Article-5687" OrderedQuantity="2" DispensedQuantity="0" PaidQuantity="0" Price="8.90" VAT=19.0"/> </ShoppingCart> <UpdateResult Status="NotUpdated" Description="Coupon not supported at self-checkout. Please contact a staff member." /> </ShoppingCartUpdateResponse> </WWKS> </pre>	

Ambiguous barcode scanned

In case the PIS needs additional information to handle an update request, it may answer with an `UpdateResult.Status="Query"` and provide multiple answer options from which the user of the device may choose. The device presents the answer options to the user and sends another update request with the same update element as before and an additional `QueryAnswer` element that contains the selected answer option.

Examples:

- A scanned barcode value may be a valid PZN and a valid EAN8 which identify two different articles sold by the pharmacy. The PIS asks which of the articles has been scanned.
- A scanned barcode may be either an article or a loyalty card. The PIS asks which of the two has been scanned.
- A loyalty card is used shared by all members of a family. The PIS queries which member of the family is in front of the machine.

<pre> <WWKS Version="2.0" TimeStamp="2013-04-16T11:14:00Z"> <ShoppingCartUpdateRequest Id="1104" Source="701" Destination="100" ShoppingCartId="SC-2345"> <BarcodeScanned Barcode="8934587345"> </ShoppingCartUpdateRequest> </WWKS> </pre>	
---	--

```

<WWKS Version="2.0" TimeStamp="2013-04-16T11:14:00Z">
  <ShoppingCartUpdateResponse Id="1104" Source="100"
    Destination="701">
    <ShoppingCart Id="SC-2345" Status="Active" SalesPointId="701"
      Currency="EUR">
      <ShoppingCartItem ArticleId="1234" Name="Article-1234"
        OrderedQuantity="1" DispensedQuantity="0" PaidQuantity="0"
        Price="9.90" VAT=19.0"/>
      </ShoppingCartItem>
    </ShoppingCart>
    <UpdateResult Status="Query" Description="Ambiguous barcode. What kind
of item did you scan?">
      <Query Id="1" Text="Superdrug 1435"/>
      <Query Id="3" Text="Loyalty card"/>
    </UpdateResult>
  </ShoppingCartUpdateResponse>
</WWKS>

```

```

<WWKS Version="2.0" TimeStamp="2013-04-16T11:14:00Z">
  <ShoppingCartUpdateRequest Id="1104" Source="701"
    Destination="100" ShoppingCartId="SC-2345">
    <BarcodeScanned Barcode="8934587345">
    <QueryAnswer Id="3"/>
  </ShoppingCartUpdateRequest>
</WWKS>

```

```

<WWKS Version="2.0" TimeStamp="2013-04-16T11:14:00Z">
  <ShoppingCartUpdateResponse Id="1104" Source="100"
    Destination="701">
    <ShoppingCart Id="SC-2345" Status="Active" SalesPointId="701"
      Currency="EUR" CustomerId="87589" >
      <ShoppingCartItem ArticleId="1234" Name="Article-1234"
        OrderedQuantity="1" DispensedQuantity="0" PaidQuantity="0"
        Price="9.90" VAT=19.0"/>
      </ShoppingCartItem>
    </ShoppingCart>
    <UpdateResult Status="Updated"/>
  </ShoppingCartUpdateResponse>
</WWKS>

```


Maximum reaction times

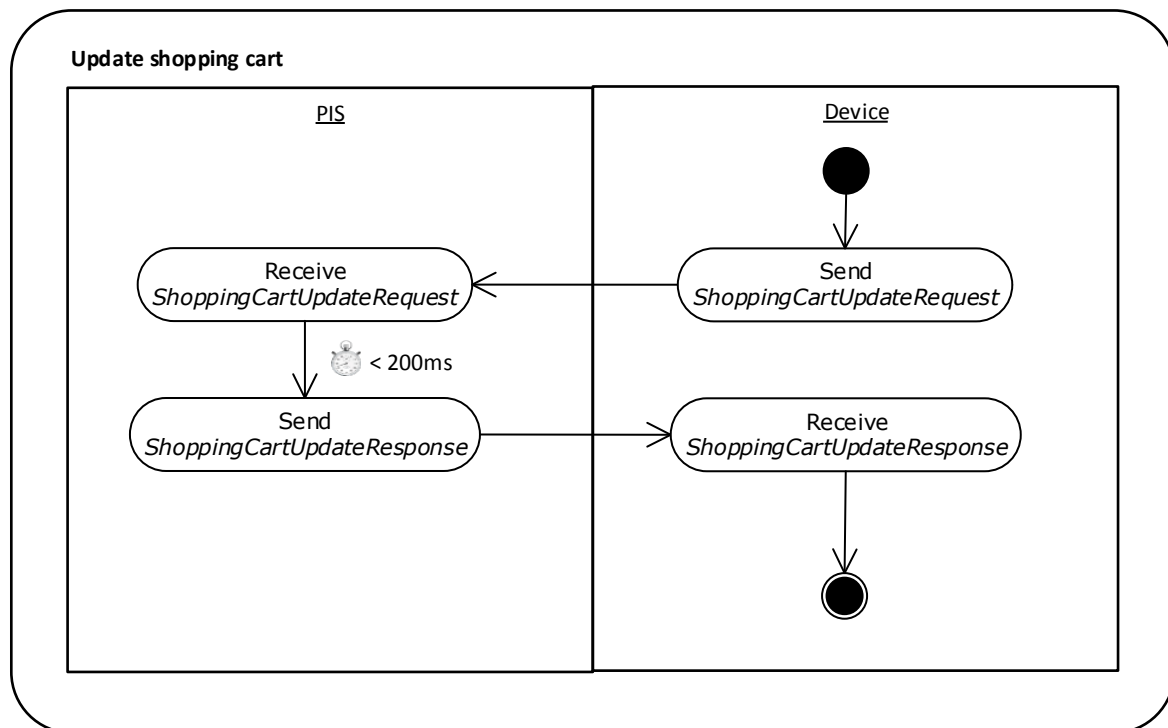
Processing step	Maximum Duration	Reaction of other side if no message is sent in time
Receiving <code>ShoppingCartUpdateRequest</code> until sending <code>ShoppingCartUpdateResponse</code>	1s	Assume that modification has been denied.

7.9.4 Pick shopping cart

Please refer to full WWKS 2+ specification, not required for self-checkout terminals.

7.9.5 Terminate handling of shopping cart at a device**Purpose**

Information by the device to the PIS that it has finished the current action on the sales transaction and will not process it any further now.

Default procedure

To inform the PIS that a device has finished working on the sales transaction, it sends a `ShoppingCartUpdateRequest` with update element `Close` to the PIS.

Deprecated: In previous versions of WWKS2+ the device just set an empty `SalesPointId` to indicate closing of a transaction.

A rejection of the update request by the PIS does not mean the device has to continue processing the transaction.

Example

```
<WWKS Version="2.0" TimeStamp="2013-04-16T11:14:00Z">
  <ShoppingCartUpdateRequest Id="1110" Source="701"
    Destination="100" ShoppingCartId="SC-2345">
    <Close/>
  </ShoppingCartUpdateRequest>
</WWKS>
```

```
<WWKS Version="2.0" TimeStamp="2013-04-16T11:14:00Z">
  <ShoppingCartUpdateResponse Id="1110" Source="100"
    Destination="701">
    <ShoppingCart Id="SC-2345" Status="Finished" SalesPointId="">
      <ShoppingCartItem ArticleId="5678" OrderedQuantity="2" .../>
    </ShoppingCart>
    <UpdateResult Status="Updated"/>
  </ShoppingCartUpdateResponse>
</WWKS>
```

Maximum reaction times

Processing step	Maximum Duration	Reaction of other side if no message is sent in time
Sending ShoppingCartUpdateRequest until receipt of ShoppingCartUpdateResponse	200ms	none

7.9.6 Pay shopping cart

7.9.6.1 Card terminal handling by device

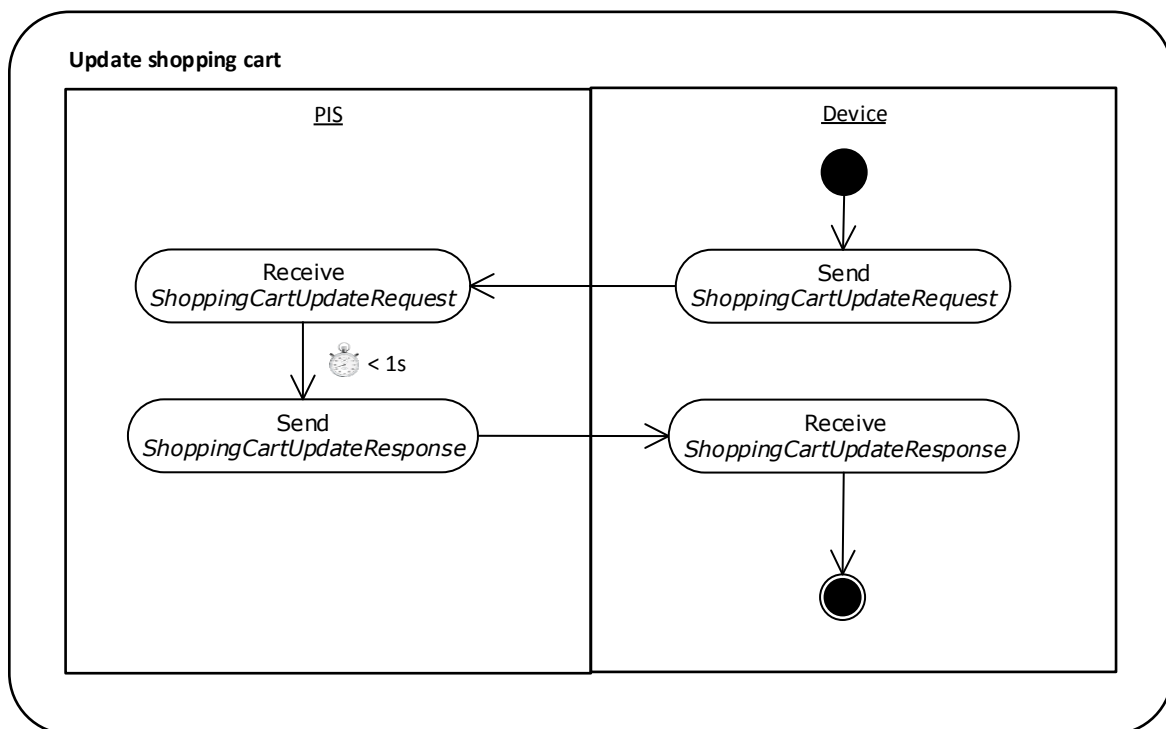
Purpose

A device tells the PIS that a shopping cart has been completely paid. It sends a `ShoppingCartUpdateRequest` that contains a `Payment` update element which includes a `PaymentItem` element that give details about the transaction and the amount paid.

Deprectated: Adjusted `PaidQuantity` for the affected line items define which parts of the shopping cart have been paid.

The payment of a sales transaction does not include picking or delivery of items to the patient.

Default procedure



Checking of already paid part quantities

It is the responsibility of the device to check if the shopping cart has been paid already partly (`PaymentItems` already exist) and not to charge these quantities again to the patient.

Payment of shopping cart

To indicate payment of the shopping cart, the device sends a `ShoppingCartUpdateRequest` with `Payment` update element including a `PaymentItem` element with `Given` and `Currency` attributes filled. Though the shopping cart itself may contain multiple `PaymentItem` elements, each update request may contain only a single `PaymentItem`.

Deprecated: If possible to tell, the device increases the `PaidQuantity` attribute of the shopping cart line items to indicate which packs have been paid.

In case the PIS accepts the payment, it adds the `PaymentItem` element to the shopping cart and fills the `Paid` attribute of it. In case the payment was done in a different currency than the main currency of the shopping cart and the PIS support currency conversion, it converts the `Given` amount into the main currency before filling the `Paid` attribute.

The PIS finally confirms the payment with a `ShoppingCartUpdateResponse` with `<UpdateResult Status="Updated"/>` and the updated state of the shopping cart.

```
<WWKS Version="2.0" TimeStamp="2013-04-16T11:14:00Z">
  <ShoppingCartUpdateRequest Id="1110" Source="701"
    Destination="100" ShoppingCartId="SC-2345">
    <Payment>
      <PaymentItem Given="43.75" Currency="EUR"/>
    </Payment>
  </ShoppingCartUpdateRequest>
</WWKS>
```

```
<WWKS Version="2.0" TimeStamp="2013-04-16T11:14:00Z">
  <ShoppingCartUpdateResponse Id="1110" Source="100"
    Destination="701">
    <ShoppingCart Id="SC-2345" Status="Active" Currency="EUR"
      SalesPointId="701">
      <ShoppingCartItem ArticleId="1234" OrderedQuantity="1" .../>
      <ShoppingCartItem ArticleId="5678" OrderedQuantity="2" .../>
      <PaymentItem Given="43.75" Currency="EUR" Paid="43.75"/>
    </ShoppingCart>
    <UpdateResult Status="Updated"/>
  </ShoppingCartUpdateResponse>
</WWKS>
```

Payment rejection by the PIS

If the PIS rejects the payment, the device has to cancel the payment and return any money already collected to the patient.

The PIS sends a `ShoppingCartUpdateResponse` with `<UpdateResult Status="NotUpdated"/>` and updated shopping cart state:

```
<WWKS Version="2.0" TimeStamp="2013-04-16T11:14:00Z">
  <ShoppingCartUpdateRequest Id="1110" Source="701"
    Destination="100" ShoppingCartId="SC-2345">
    <Payment>
      <PaymentItem Given="100.00" Currency="BonusPoints"/>
    </Payment>
  </ShoppingCartUpdateRequest>
</WWKS>
```

```
<WWKS Version="2.0" TimeStamp="2013-04-16T11:14:00Z">
  <ShoppingCartUpdateResponse Id="1110" Source="100"
    Destination="701" Currency="EUR">
    <ShoppingCart Id="SC-2345" Status="Active" SalesPointId="">
      <ShoppingCartItem ArticleId="1234" OrderedQuantity="1" .../>
      <ShoppingCartItem ArticleId="5678" OrderedQuantity="2" .../>
      <!-- PaymentItem Given="100.00" Currency="BonusPoints" / -->
    </ShoppingCart>
    <UpdateResult Status="NotUpdated"
      Description="Payment with BonusPoints not supported."/>
  </ShoppingCartUpdateResponse>
</WWKS>
```

Payment transaction details

The device may add details about the transaction (e.g. as reported by the card payment device) as `TransactionDetail` sub elements to the `PaymentItem` element.

As the details provided depend on the payment method and the payment device, WWKS 2+ does not predefine any identifiers for transaction details. In case a further processing beyond logging is required, PIS and device vendors have to agree on the identifiers used.

In case the payment device provides all transaction details in just one value, just this `TransactionDetail` element is provided.

```
<WWKS Version="2.0" TimeStamp="2013-04-16T11:14:00Z">
  <ShoppingCartUpdateRequest Id="1110" Source="701"
```

```

    Destination="100" ShoppingCartId="SC-2345">
    <Payment>
      <PaymentItem Given="43.75" Currency="EUR">
        <TransactionDetail Id="Method" Value="Card Payment Visa"/>
        <TransactionDetail Id="CardNumber" Value="1234xxxxxxxx34"/>
        <TransactionDetail Id="CardValidUntil" Value="12/23"/>
        <TransactionDetail Id="TerminalId" Value="71014925"/>
        <TransactionDetail Id="TerminalId" Value="71014925"/>
        <TransactionDetail Id="TANo" Value="972297"/>
        <TransactionDetail Id="TransactionNumber" Value="2012"/>
        <TransactionDetail Id="Receipt" Value="03/8219/2454"/>
        <TransactionDetail Id="ACT-Code" Value="0000000A"/>
        <TransactionDetail Id="VU-Nummer" Value="4556705160"/>
        <TransactionDetail Id="Amount" Value="43.75 EUR"/>
        <TransactionDetail Id="Success" Value="Payment made"/>
      </PaymentItem>
    </Payment>
  </ShoppingCartUpdateRequest>
</WWKS>

```

Deprecated usage

Former releases of WWKS 2+ did not define the `PaymentItem` element of the shopping cart and used just the `PaidQuantity` line item attribute to indicate which parts of the shopping cart have been paid. In addition, no main currency for the shopping cart was defined.

It is recommend to always use the `PaymentItem` elements as these allow partial payments that do not match a selection of packs, payment in other currencies and transfer of transaction details. Filling the `PaidQuantity` attribute is optional if the `PaymentItem` element is used.

Maximum reaction times

Processing step	Maximum Duration	Reaction of other side if no message is sent in time
Send out ShoppingCartUpdateRequest until receipt of ShoppingCartUpdateResponse	1s	Assume that payment has been rejected.

7.9.6.2 Card terminal handling by PIS

Purpose

In difference to the generic Pay shopping cart use case, the PIS may take over the handling of the card terminal. The device still manages the shopping cart and guides the user through the payment process.

Default procedure

At the moment the user should perform the actual payment, the device sends a `ShoppingCartUpdateRequest` with a `PaymentRequest` update element.

As part of the request, the device provides the number of characters its receipt printer is able to print within one row, so that the PIS may provide this information to the terminal to receive proper formatted transaction details.

The PIS triggers the card terminal. After the transaction is finished successfully, the PIS returns a `ShoppingCartUpdateResponse` that contains the new `PaymentItem` including the transaction details. In case the payment device provides all transaction details in just one value, just one `TransactionDetail` element is provided.

```
<WWKS Version="2.0" TimeStamp="2013-04-16T11:14:00Z">
  <ShoppingCartUpdateRequest Id="1110" Source="701"
    Destination="100" ShoppingCartId="SC-2345">
    <PaymentRequest ReceiptWidth="80"/>
  </ShoppingCartUpdateRequest>
</WWKS>
```

```
<WWKS Version="2.0" TimeStamp="2013-04-16T11:14:00Z">
  <ShoppingCartUpdateResponse Id="1110" Source="701"
    Destination="100">
    <ShoppingCart Id="SC-2345" Status="Active" Currency="EUR"
      SalesPointId="701">
      <ShoppingCartItem ArticleId="1234" OrderedQuantity="1"
        PaidQuantity="1" Price="34.95" VAT="19.0".../>
      <ShoppingCartItem ArticleId="5678" OrderedQuantity="2"
        PaidQuantity="2" Price="8.80" VAT="19.0".../>
      <PaymentItem Given="43.75" Currency="EUR">
        <TransactionDetail Id="Method" Value="Card Payment Visa"/>
        <TransactionDetail Id="CardNumber" Value="12xxxxxxxx34"/>
        <TransactionDetail Id="CardValidUntil" Value="12/23"/>
        <TransactionDetail Id="TerminalId" Value="71014925"/>
        <TransactionDetail Id="TerminalId" Value="71014925"/>
        <TransactionDetail Id="TANo" Value="972297"/>
        <TransactionDetail Id="TransactionNumber" Value="2012"/>
        <TransactionDetail Id="Receipt" Value="03/8219/2454"/>
        <TransactionDetail Id="ACT-Code" Value="0000000A"/>
        <TransactionDetail Id="VU-Nummer" Value="4556705160"/>
        <TransactionDetail Id="Amount" Value="43.75 EUR"/>
        <TransactionDetail Id="Success" Value="Payment made"/>
      </PaymentItem>
    </ShoppingCart>
    <UpdateResult Status="Updated"/>
  </ShoppingCartUpdateResponse>
</WWKS>
```

```
</ShoppingCartUpdateResponse>
</WWKS>
```

Payment not successful

If the payment wasn't successful, e.g. because of a terminal error or user abortion, the PIS sends a `ShoppingCartUpdateResponse` with `UpdateResult Status="NotUpdated"` and gives an error reason in the `Description` attribute.

```
<WWKS Version="2.0" TimeStamp="2013-04-16T11:14:00Z">
  <ShoppingCartUpdateRequest Id="1110" Source="701"
    Destination="100" ShoppingCartId="SC-2345">
    <PaymentRequest ReceiptWidth="80"/>
  </ShoppingCartUpdateRequest>
</WWKS>
```

```
<WWKS Version="2.0" TimeStamp="2013-04-16T11:14:00Z">
  <ShoppingCartUpdateResponse Id="1110" Source="701"
    Destination="100">
    <ShoppingCart Id="SC-2345" Status="Active" Currency="EUR"
      SalesPointId="701">
      <ShoppingCartItem ArticleId="1234" OrderedQuantity="1"
        PaidQuantity="0" Price="34.95" VAT="19.0".../>
      <ShoppingCartItem ArticleId="5678" OrderedQuantity="2"
        PaidQuantity="0" Price="8.80" VAT="19.0".../>
    </ShoppingCart>
    <UpdateResult Status="NotUpdated" Description="Aborted by user"/>
  </ShoppingCartUpdateResponse>
</WWKS>
```

Maximum reaction times

Processing step	Maximum Duration	Reaction of other side if no message is sent in time
Send out <code>ShoppingCartUpdateRequest</code> Action="PaymentRequest" until receipt of <code>ShoppingCartUpdateResponse</code>	120s	Assume that payment has been aborted.

7.9.6.3 Receipt layout by PIS

Purpose

To unify the layout of receipts printed within the pharmacy, the PIS may provide the receipt as printable layout when a payment has been performed.

Default procedure

In the `ShoppingCartUpdateRequest` with update element `PaymentRequest` or `Payment`, the device fills the attribute `ReceiptWidth` to inform the PIS about the paper width available to print the receipt. This value is only relevant if the PIS returns a text based receipt content but not a picture or PDF.

In the `ShoppingCardUpdateResponse` answering a request with update element `PaymentRequest` or `Payment`, the PIS sends a `Receipt` sub element of the `PaymentItem` element.

The `Receipt` element contains the receipt data as raw data content according to the format and encoding specified.

```
<WWKS Version="2.0" TimeStamp="2013-04-16T11:14:00Z">
  <ShoppingCartUpdateRequest Id="1110" Source="701"
    Destination="100" ShoppingCartId="SC-2345">
    <Payment ReceiptWidth="80">
      <PaymentItem Given="43.75" Currency="EUR"/>
    </Payment>
  </ShoppingCartUpdateRequest>
</WWKS>
```

```
<WWKS Version="2.0" TimeStamp="2013-04-16T11:14:00Z">
  <ShoppingCardUpdateResponse Id="1110" Source="701"
    Destination="100">
    <ShoppingCart Id="SC-2345" Status="Active" Currency="EUR"
      SalesPointId="701">
      <ShoppingCartItem ArticleId="1234" OrderedQuantity="1"
        PaidQuantity="1" Price="34.95" VAT="19.0".../>
      <ShoppingCartItem ArticleId="5678" OrderedQuantity="2"
        PaidQuantity="2" Price="8.80" VAT="19.0".../>
      <PaymentItem Given="43.75" Currency="EUR">
        <TransactionDetail Id="Method" Value="Card Payment Visa"/>
        <TransactionDetail Id="CardNumber" Value="12xxxxxxxxxx34"/>
        ...
        <Receipt Format="JPG" Encoding="Base64">
          /9j/4AAQSkZJRgABAQEAYABgAAD/4QBoRXhpZgAATU0A...
        </Receipt>
      </PaymentItem>
    </ShoppingCart>
    <UpdateResult Status="Updated"/>
  </ShoppingCardUpdateResponse>
</WWKS>
```

Multiple receipts

The PIS may provide more than one receipt (part). The device will print all of them in the order provided.

```
<WWKS Version="2.0" TimeStamp="2013-04-16T11:14:00Z">
  <ShoppingCartUpdateResponse Id="1110" Source="701"
    Destination="100">
    <ShoppingCart Id="SC-2345" Status="Active" Currency="EUR"
      SalesPointId="701">
      <ShoppingCartItem ArticleId="1234" OrderedQuantity="1"
        PaidQuantity="1" Price="34.95" VAT="19.0".../>
      <ShoppingCartItem ArticleId="5678" OrderedQuantity="2"
        PaidQuantity="2" Price="8.80" VAT="19.0".../>
      <PaymentItem Given="43.75" Currency="EUR">
        <TransactionDetail Id="Method" Value="Card Payment Visa"/>
        <TransactionDetail Id="CardNumber" Value="12xxxxxxxxxx34"/>
        ...
        <Receipt Format="JPG" Encoding="Base64">
          /9j/4AAQSkZJRgABAQEAYABgAAD/4QBoRXhpZgAATU0A...
        </Receipt>
        <Receipt Format="PDF" Encoding="Base64">
          JVBERi0xLjUKJcKlwrXCtcK1CjEgMCBvYmoKPDwvVHlw...
        </Receipt>
      </PaymentItem>
    </ShoppingCart>
    <UpdateResult Status="Updated"/>
  </ShoppingCartUpdateResponse>
</WWKS>
```

7.9.6.4 Partial payment and payment with bonus points

Purpose

Support of this use case allows the customer to pay just parts of the shopping cart and/or use other currencies than the main currency of the shopping cart like bonus points to the shopping cart partly or fully.

Capability

Partial payment and payment using alternative currencies like bonus points requires that PIS and device support the capability `PartialPayment`.

Partial payment

In case the patient does not fully pay the shopping cart, the device sends a `PaymentItem` element that does not cover the full sum of all shopping cart items.

Deprecated: If applicable, the device increases the `PaidQuantity` just for paid packs. If the partial payment does not match a selection of packs from the shopping cart, the device does not change the `PaidQuantity` attributes.

Example:

The customer pays just 10 EUR of the total sum of 43.75 EUR:

```
<WWKS Version="2.0" TimeStamp="2013-04-16T11:14:00Z">
  <ShoppingCartUpdateRequest Id="1110" Source="701"
    Destination="100" ShoppingCartId="SC-2345">
    <Payment>
      <PaymentItem Given="10.00" Currency="EUR"/>
    </Payment>
  </ShoppingCartUpdateRequest>
</WWKS>
```

Example 2:

At a later point in time, the customer pays the remaining amount of items. The first `PaymentItem` with 10 EUR is still in the shopping cart.

```
<WWKS Version="2.0" TimeStamp="2013-04-16T11:14:00Z">
  <ShoppingCartUpdateRequest Id="1110" Source="701"
    Destination="100" ShoppingCartId="SC-2345">
    <Payment>
      <PaymentItem Given="33.75" Currency="EUR"/>
    </Payment>
  </ShoppingCartUpdateRequest>
</WWKS>
```

Payment in different currency or with bonus points

In case the customer pays using a different currency than the main currency of the shopping cart, the set the `Currency` attribute of the `PaymentItem` attribute.

In case of a non-standard currency like bonus points, PIS and device vendor have to agree on the value for the currency attribute.

After receiving the `ShoppingCartUpdateRequest` the PIS converts the amount given into the main currency of the shopping cart and fills the `Paid` attribute of the `PaymentItem` element. The shopping cart is fully paid if and only if the sum of the `Paid` attributes equals the sum of all line item price attributes. Line items with a currency different to the main currency are not taken into account when calculating the amount to pay.

```
<WWKS Version="2.0" TimeStamp="2013-04-16T11:14:00Z">
  <ShoppingCartUpdateRequest Id="1110" Source="701"
    Destination="100" ShoppingCartId="SC-2345">
    <Payment>
      <PaymentItem Given="100.00" Currency="BonusPoints"/>
    </Payment>
  </ShoppingCartUpdateRequest>
</WWKS>
```

```
<WWKS Version="2.0" TimeStamp="2013-04-16T11:14:00Z">
  <ShoppingCartUpdateResponse Id="1110" Source="100"
    Destination="701">
    <ShoppingCart Id="SC-2345" Status="Active" Currency="EUR"
      SalesPointId="">
      <ShoppingCartItem ArticleId="1234" OrderedQuantity="1"
        PaidQuantity="1" .../>
      <ShoppingCartItem ArticleId="5678" OrderedQuantity="2"
        PaidQuantity="2" .../>
      <PaymentItem Given="100.00" Currency="BonusPoints" Paid="8.43"/>
    </ShoppingCart>
    <UpdateResult Status="Updated"/>
  </ShoppingCartUpdateResponse>
</WWKS>
```

7.9.7 Deliver shopping cart to patient

Please refer to full WWKS 2+ specification, not required for pick-up terminals.

7.9.8 Transfer shopping cart to another sales point

To be defined

7.9.9 Inform about shopping cart modifications

Usually not required for self-checkout terminals as they handle a shopping cart exclusively. Please refer to full WWKS 2+ specification.

7.9.10 Query historic, already closed shopping carts

Please refer to full WWKS 2+ specification, not required for self-checkout terminals.

7.9.11 Select item

Please refer to full WWKS 2+ specification, not required for self-checkout terminals.

7.9.12 Shopping cart with item master data

Purpose

A device may display additional item information that are not part of a shopping cart. The PIS may transmit them embedded in the shopping part to avoid additional queyring of this data by the device.

Default procedure

When providing shopping cart information in a `ShoppingCartResponse` or `ShoppingCartUpdateResponse`, the PIS embeddeds `Article` elements as sub element of the `ShoppingCartItem`.

It depends on the device with `Article` attributes and sub elements are handled.

The devices has to be prepared to handle the situation that the PIS does not provide an attribute or sub element it expects.

Currently, the Rowa Self-Checkout is the only Rowa device handling `Article` elements in the `ShoppingCart`.

Example

```
<WWKS Version="2.0" TimeStamp="2013-04-16T11:14:00Z">
  <ShoppingCartUpdateResponse Id="1104" Source="999" Destination="100"
    Action="Order">
    <ShoppingCart Id="SC-2345">
      <ShoppingCartItem ArticleId="1111" Name="Article-1111"
        OrderedQuantity="5"/>
      <ShoppingCartItem ArticleId="1111" Name="Article-1111"
        OrderedQuantity="1" DispensedQuantity="0" PaidQuantity="0"
        Price="15.00" VAT="19.0">
        <Article Id="1111">
          <PriceInformation Category="Standard" Description="List price"
            Quantity="1" Price="20.00" BasePrice="10.00"
            BasePriceUnit="100ml" VAT="19"/>
          <PriceInformation Category="Offer" Description="Offer"
            Quantity="1" Price="15.00" BasePrice="7.50"
            BasePriceUnit="100ml" VAT="19"/>
          <Tag Value="AgeVerification"/>
        </Article>
      </ShoppingCartItem>
      <ShoppingCartItem ArticleId="2222" Name="Article-2222"
        OrderedQuantity="4" Price="18.95" VAT="19.0"/>
      <Article Id="2222">
        <PriceInformation Category="Standard" Description="List price"
          Quantity="1" Price="18.95" BasePrice="18.95" BasePriceUnit="10"
          VAT="19"/>
        <Tag Value="Consultation"/>
      </Article>
    </ShoppingCart>
  </ShoppingCartUpdateResponse>
</WWKS>
```

```
</Article>
  <ShoppingCartItem/>
</ShoppingCart>
  <UpdateResult Status="Updated" Description="Item has been added."/>
</ShoppingCartUpdateResponse>
</WWKS>
```

8 Data structures

WWKS2 uses common data structures for all messages.

Most sub-elements and attributes of a data structure are optional, but are mandatory in certain use cases. This is specified in the description of the respective use cases.

If an attribute is not specified, but is expected from the opposite side, then it uses the respective defined default value (see below) of the attribute instead.

8.1 Data types

Messages and data structures use these elemental data types:

Data type	Description															
Tag	XML tag															
String	<p>Character string (corresponding to the data type <i>string</i> of the W3C specification for the XML scheme)</p> <p>The permissible value ranges are defined as follows: <code>#x9 #xA [#x20-#xD7FF] [#xE000-#xFFFD] [#x10000-#x10FFFF]</code> and each Unicode character apart from the surrogate blocks <i>FFFE</i> and <i>FFFF</i>, as specified in the standard ISO/IEC 10646.</p> <p>All control characters (e.g. <code>#x1D</code>) not specified in the ISO standard must be coded with <code>\x00</code>. The zeros represent the respective HEX code. Example: <code>\x1D</code>.</p> <p>These characters must be substituted:</p> <table><tr><td>&</td><td>by</td><td>&amp;</td></tr><tr><td><</td><td>by</td><td>&lt;</td></tr><tr><td>></td><td>by</td><td>&gt;</td></tr><tr><td>"</td><td>by</td><td>&quot;</td></tr><tr><td>'</td><td>by</td><td>&apos;</td></tr></table>	&	by	&	<	by	<	>	by	>	"	by	"	'	by	'
&	by	&														
<	by	<														
>	by	>														
"	by	"														
'	by	'														
String64	A <i>String</i> with a maximum length of 64 characters.															
FormattedString	<p>A <i>String</i> that may contain format information.</p> <table><tr><td>[br]</td><td>New line</td></tr><tr><td>[tab]</td><td>Indent</td></tr><tr><td>[u]</td><td>Start/end underline</td></tr><tr><td>[b]</td><td>Start/end bold text</td></tr><tr><td>[i]</td><td>Start/end italic text</td></tr></table>	[br]	New line	[tab]	Indent	[u]	Start/end underline	[b]	Start/end bold text	[i]	Start/end italic text					
[br]	New line															
[tab]	Indent															
[u]	Start/end underline															
[b]	Start/end bold text															
[i]	Start/end italic text															

Integer 32-bit	<p>32-bit integer (corresponding to the data type <i>int</i> of the W3C specification for the XML scheme)</p> <p>This data type has a syntax comprising a sequence of integers (#x30-#x39).</p> <p>Thousands separators and decimal points are not allowed.</p> <p>The permitted value range covers the numbers between -2147483648 and 2147483647.</p>
Integer 64-bit	<p>64-bit integer (corresponding to the data type <i>long</i> of the W3C specification for the XML scheme)</p> <p>This data type has a syntax comprising a sequence of integers (#x30-#x39).</p> <p>Thousands separators and decimal points are not allowed.</p> <p>The permitted value range covers the numbers between -9223372036854775808 and 9223372036854775807.</p>
Decimal	<p>Decimal number with any precision (corresponds to the data type <i>float</i> in the W3C specification for the XML schema) with at least 18 significant digits.</p> <p>This data type has a lexical representation consisting of a sequence of integers (#x30-#x39).</p> <p>Thousands of points are not allowed, decimal points are allowed.</p>
Boolean	<p>One of the statements <i>True</i> or <i>False</i>.</p> <p>This is a deviation from standard XML which defines lower case values!</p>
Date	<p>Date in the format YYYY-MM-TT</p>

In case an attribute of a message is defined as optional and it is not send in this message, the recipient uses the default value defined in tables in the following sections. In some use cases different default values might be used, this is noted in the use case description.

In some cases, the types described here are restricted if they are used in attributes (e.g. 32-bit integer > 0). This is noted at the relevant locations.

8.2 Stock

8.2.1 Article

The use of article elements and the ArticleInfoRequest/-Response and ArticlePriceRequest/-Response is deprecated for self-checkout terminals. Please embed them in ShoppingCartItem elements instead.

For a list of all article attribute, please refer to full WWKS 2+ specification. This document just lists the attributes relevant for self-checkout.

An article describes the properties of an item sold in the pharmacy that do not or just rarely change.

Structure

```
<Article>
</Article>
```

There are several optional sub-elements for an article, which are described in the following sections

Element	Data type	Description		Handled in Shopping Cart
Article	Tag	Article information. This element may occur multiple times.		
Attributes	Data Type	Description and values	Default	
Id	String64	ID of the item. This can be the barcode on the original manufacturer pack or another value. The PIS defines the content of the article Id.	None	SCO
Name	String	Item name	""	SCO
DosageForm	String	Dosage form of the item.	""	SCO
PackingUnit	String	Packaging unit of the item.	""	SCO
RequiresFridge	Boolean	Flag that indicates if the item has to be stored in a cooled environment (2°C - 8°C) ("True").	False	
MaxSubItemQuantity	Integer 32-bit >0	Number of unit doses (e.g. tablets, ampules, milliliter) in a full pack of the item. A value of „0“ means that the number is unknown.	0	

PatientInformationLeaflet	String	Content of the article patient leaflet.	""	
Description	FormattedString	Detailed article description for display to a patient on a digital screen or a shop system.	""	

8.2.1.1 Product-/Barcodes

Please refer to full WWKS 2+ specification, not required for self-checkout terminals.

8.2.1.2 Prices

An article may have one or multiple prices. The attribute `Category` defines the type of the price and when and how it has to be used.

Structure

A price information is always a sub-element of an article.

```
<Article>
  <PriceInformation>
  </PriceInformation>
</Article>
```

Element	Data type	Description	
Price Information	Tag	Price information follows. May appear multiple time.	
Attribute	Data type	Description and values	Default
Category	String	<p>The type of a price.</p> <p>WWKS 2+ defines some standard categories as listed below. The PIS may define and transmit additional categories.</p> <p>„RRP“: recommended retail price of the article manufacturer,</p> <p>„Standard“: default price without discount usually used in the pharmacy. The price that is shown on a price tag in the pharmacy rooms.</p> <p>„Offer“: The currently special offer price. Transmit only if the device shall display the price in an emphasized way, e.g. in a different color or comparing it to the default price.</p> <p>“Member”: price that is valid for a selected customer group only and that is displayed in parallel to standard prices on price tags</p>	„Standard“

Description	String	Human readable description of the price category.	„"
Quantity	Integer 32-bit >0	Minimum number of packs in a shopping cart line needed to apply the price.	1
Price	Decimal >0	The price (including VAT) for one pack of the article.	0.00
BasePrice	Decimal >0	The price for a defined amount of doses (see BasePriceUnit) of the article. Used to make different pack sizes comparable and mandatory for price tags in some countries.	0.00
BasePriceUnit	String	The unit for which the base price is defined (e.g. "100 ml")	""
VAT	Decimal >0	The value added tax rate that is included in the price.	0.00
Currency	String	Currency code as defined by ISO 4217 (e.g. „EUR") in which the price is defined. Default value is "" meaning undefined currency.	""

8.2.1.3 Tags

An article may be assigned an unlimited number of tags that a device may use to filter or group a list of articles or to trigger special handling actions.

Some tags may have special meanings depending on the context. For details please refer to the use case and message descriptions.

Structure

A tag is always a sub-element of an article.

```
<Article>
  <Tag>
  </Tag>
</Article>
```

Element	Data type	Description	
Tag	Tag	Article tag follows. May appear multiple time.	
Attribute	Data type	Description and values	Default
Value	String	Tag value. Predefined values, the PIS may transfer additional values: "AgeVerification": Item requires age verification before handing over to customer. "Consultation": Item requires consultation by pharmacy staff before handing over to customer.	" "

8.2.1.4 Article Relations

Please refer to full WWKS 2+ specification, currently not supported for self-checkout terminals.

8.2.2 Packs

A pack describes the properties of a single sales unit of an article. It covers only properties that are not already part of the article data structure.

Please refer to full WWKS 2+ specification, not required for self-checkout terminals.

8.3 Good receipt

Please refer to full WWKS 2+ specification, not required for self-checkout terminals.

8.4 Sales Transaction (Shopping cart)

The essential, information-carrying element for sales transactions is the `ShoppingCart`.

It contains the basic information about the sales transaction and a list of items contained in the shopping cart for the transaction. In addition may contain a list of information items that are not related to a line item but should be displayed or printed on the receipt of the transaction.

The current state of the sales transaction is always defined by the pharmacy IT system. The device may just ask for modifications and depend on the IT system to confirm them.

In the case of change requests for sales transactions using the `ShoppingCartUpdateRequest`, only similar changes can be transferred in a request and the type of is transferred in the `Action` attribute. The column "Update Action" of the following tables defines the corresponding update action for each element of the sales transaction. See 12.2.1 `ShoppingCartUpdateRequest`.

Structure

```
<ShoppingCart>
  <ShoppingCartItem>
    <Article>
      <PriceInformation/>
      <Tag/>
    </Article>
  </ShoppingCartItem>
  <ShoppingCartInfo/>
  <PaymentItem/>
    <TransactionDetail/>
    <Receipt/>
  </PaymentItem>
</ShoppingCart>
```

Elements

Element	Data type	Description	
Shopping Cart	Tag	Sales transaction details.	
Attribute	Data type	Description and value	Default
Id	String64	Id of the shopping cart.	„"
Status	String	Current status of the sales transaction. Possible values:	„Active"

		<p>"Active", if the order process is ongoing/being processed.</p> <p>„Consultation“, if a consultation is required before the shopping cart may be dispensed to the patient.</p> <p>"Finished", if the sales transaction has been closed for now. It may be reopened at a later point in time.</p> <p>"Discarded", if the order process has been discarded.</p> <p>„NotActive“, if the transaction is currently not available for processing, e.g. because it is assigned to a different point of sale.</p>	
StatusMessage	String	Details of status from the PIS to be displayed to the customer.	„"
SalesPointId	String64	<p>ID of the point of sale where the transaction is currently handled.</p> <p>Empty, if no point of sale is currently assigned.</p>	„"
SalesPersonId	String64	ID of the responsible salesperson.	„"
CustomerId	String64	ID of the patient.	„"
PickupId	String64	Code to request dispensing of the shopping cart already sent to a patient.	„"
Currency	String	<p>Currency used for this shopping cart.</p> <p>ISO 4217 currency code, e.g. 'EUR'</p>	„"
DisplayLanguage	String	<p>Display language of the device.</p> <p>ISO-639-1 code, e.g. "en".</p> <p>Empty if undefined.</p>	""

Element	Data type	Description	Default
ShoppingCartItem	Tag	Details of an item in the shopping cart. This element may occur multiple times.	
Attribute	Data type	Description and values	
ArticleId	String64	ID of the line item in the shopping cart. The same id may appear in multiple line items of the shopping cart. The ID may identify any kind of line item in the shopping cart, not necessarily just articles.	""
Name	String	Line item name to be displayed within the shopping cart. May differ from article name defined in article master data.	""
Description	Formatted Text	Additional information for the line item to be displayed by the device.	""
OrderedQuantity	Integer 32-bit ≥0	Number of packs of the article originally ordered by the patient.	0
AvailableQuantity	Integer 32-bit ≥0	Number of packs of the article on stock in the pharmacy and available for the patient to collect.	Unlimited
PaidQuantity	Integer 32-bit ≥0	<i>Deprecated, use PaymentItem element instead.</i> Number of packs of the article already paid by the patient. Only used if it is possible to tell which packs are covered by a partial payment.	0
Price	Decimal	The total price of the item line in the shopping cart including value added tax.	0.00
Currency	String	Currency of the line item price in case it deviates of the main currency of the shopping cart. ISO 4217 currency code, e.g. 'EUR'	ShoppingCart.Currency
VAT	Decimal ≥0	The value added tax rate that is included in the price.	0.0

ItemListPrice	Decimal	List price of one unit of the line item that the device may show as additional information.	0.0
ItemOffer Price	Decimal	Current offer price of one unit of the line item that the device may show as additional information.	0.0

Element	Data type	Description	Default
Article	Tag	Master data of article listed in a ShoppingCartItem that are relevant to display as detail information to a user in front of a device handling the shopping cart.	
Attribute	Data type	Description and values <i>See as well description of article data structure in chapter 8.2.1.</i>	
Id	String64	ID of the item.	None
Name	String	Item name.	""
DosageForm	String	Dosage form of the item.	""
PackingUnit	String	Packaging unit of the item.	""
Description	FormattedString	Detailed article description for display to a patient on a digital screen or a shop system.	„“

Element	Data type	Description	
PriceInformation	Tag	Generic price information of the line item follows. May appear multiple time. May not relate to the line item price.	
Attribute	Data type	Description and values	Default
Category	String	The type of a price. WWKS 2+ defines some standard categories as listed below. The PIS may define and transmit additional categories.	„Standard“

		<p>„RRP”: recommended retail price of the article manufacturer,</p> <p>„Standard”: default price without discount usually used in the pharmacy. The price that is shown on a price tag in the pharmacy rooms.</p> <p>„Offer”: The currently special offer price. Transmit only if the device shall display the price in an emphasized way, e.g. in a different color or comparing it to the default price.</p> <p>„Member”: price that is valid for a selected customer group only and that is displayed in parallel to standard prices on price tags</p>	
Description	String	Human readable description of the price category.	„
Quantity	Integer 32-bit >0	Minimum number of packs in a shopping cart line needed to apply the price.	1
Price	Decimal >0	The price (including VAT) for one pack of the article.	0.00
BasePrice	Decimal >0	The price for a defined amount of doses (see BasePriceUnit) of the article. Used to make different pack sizes comparable and mandatory for price tags in some countries.	0.00
BasePrice Unit	String	The unit for which the base price is defined (e.g. „100 ml”)	„
VAT	Decimal >0	The value added tax rate that is included in the price.	0.00
Currency	String	<p>Currency code as defined by ISO 4217 (e.g. „EUR”) in which the price is defined.</p> <p>Default value is „”, meaning undefined currency.</p>	„

Element	Data type	Description	
Tag	Tag	Article tag follows. May appear multiple time.	
Attribute	Data type	Description and values	Default
Value	String	<p>Tag value.</p> <p>Predefined values, the PIS may transfer additional values:</p> <p>"AgeVerification": Item requires age verification before handing over to customer.</p> <p>"Consultation": Item requires consultation by pharmacy staff before handing over to customer.</p>	"

Element	Data type	Description	Default
Shopping CartInfo	Tag	<p>Additional information about a shopping cart that should be displayed to the device user.</p> <p>May appear multiple times.</p>	
Attribute	Data type	Description and values	
Id	String	<p>Identifier of the display item. The device may use the identifier to define a layout where to display the item.</p> <p>WWKS 2+ predefined some values, but the PIS is free to transmit additional values.</p> <p>Predefined values: <i>To be defined.</i></p> <p>Default value is "". The device will display elements without id value as list in the order they have been transmitted in the shopping cart.</p> <p>The same Id may appear multiple times within one shopping cart.</p>	"
Value	String	Value to display.	"

Element	Data type	Description	Default
PaymentItem	Tag	Information about (partial) payments of the shopping cart	
Attribute	Data type	Description and values	
Given	Decimal	Amount paid in the currency used for the payment.	0.00
Currency	String	Currency used for the payment. ISO 4217 currency code, e.g. 'EUR' or a string identifying a non-standard currency like bonus points.	ShoppingCart.Currency
Paid	Decimal	Payment amount accepted by the PIS in the main currency of the shopping cart.	0.00

Element	Data type	Description	Default
Transaction Detail	Tag	Information about payment transaction details follow. May appear multiple times	
Attribute	Data type	Description and values	
Id	String	Identifier of the transaction detail.	""
Value	String	Value of the transaction detail	""

Element	Data type	Description	Default
Receipt	Tag	Information about receipt layout follows. May appear multiple.	
Attribute	Data type	Description and values	
Format	String	<p>Format of the data in the element.</p> <p>Possible values:</p> <p>"JPG": JPG encoded picture</p> <p>"PNG": PNG encoded picture</p> <p>"PDF": PDF encoded picture</p> <p>"ESCPOS": Epson ESC/POS³ formatted (<i>not supported by Rowa SCO yet</i>)</p> <p>"OPOS": hardware independent ESCPOS (<i>not supported by Rowa SCO yet</i>)</p> <p>"TEXT": Plain unicode text without any format information except CR/LF.</p>	"
Encoding	String	<p>Encoding used for the element data to avoid collision with XML specification.</p> <p>Possible values:</p> <p>"None": No encoding (default value)</p> <p>"Base64": Base64 encoding</p>	

³ See https://reference.epson-biz.com/modules/ref_escpos/index.php?content_id=2

Example

```

<ShoppingCart Id="SC-2345" SalesPointId="103" CustomerId="84585" Status="Active"
    Currency="EUR">
  <ShoppingCartItem ArticleId="1111" Name="Article-1111" OrderedQuantity="1"
    DispensedQuantity="0" PaidQuantity="0" Price="15.00" VAT="19.0">
    <Article Id="1111">
      <PriceInformation Category="Standard" Description="List price"
        Quantity="1" Price="20.00" BasePrice="10.00" BasePriceUnit="100ml"
        VAT="19"/>
      <PriceInformation Category="Offer" Description="Offer" Quantity="1"
        Price="15.00" BasePrice="7.50" BasePriceUnit="100ml" VAT="19"/>
      <Tag Value="AgeVerification"/>
    </Article>
  </ShoppingCartItem>
  <ShoppingCartItem ArticleId="2222" Name="Article-2222" OrderedQuantity="4"
    DispensedQuantity="4" PaidQuantity="4" Price="18.95"
    VAT="19.0"/>
    <Article Id="2222">
      <PriceInformation Category="Standard" Description="List price"
        Quantity="1" Price="18.95" BasePrice="18.95" BasePriceUnit="10"
        VAT="19"/>
      <Tag Value="Consultation"/>
    </Article>
  </ShoppingCartItem>
  <ShoppingCartItem ArticleId="556897" Name="Coupon Cosmetics"
    OrderedQuantity="1" DispensedQuantity="1" PaidQuantity="0" Price="-12.99"
    VAT="19.0"/>
  <ShoppingCartItem ArticleId="255448" Name="3x Points" OrderedQuantity="1"
    DispensedQuantity="1" PaidQuantity="0" Price="0.0" VAT=""/>
  <ShoppingCartInfo Item="BonusPointsAccountBalance" Value="532"/>
  <ShoppingCartInfo Item="BonusPointsForThisCart" Value="47"/>
  <PaymentItem Given="100" Currency="Points" Paid="10.00"/>
  <PaymentItem Given="8.95" Currency="EUR" Paid="8.95">
    <TransactionDetail Id="Method" Value="Card Payment Visa"/>
    <TransactionDetail Id="CardNumber" Value="12xxxxxxxxxx34"/>
    <Receipt Format="JPG" Encoding="Base64">
      /9j/4AAQSkZJRgABAQEAYABgAAD/4QBoRXhpZgAATU0A...
    </Receipt>
  </PaymentItem>
</ShoppingCart>

```

8.5 Label

Please refer to full WWKS 2+ specification, not required for self-checkout terminals.

8.6 Totes

Please refer to full WWKS 2+ specification, not required for self-checkout terminals.

8.7 Customer

To be defined.

8.8 Stock Location

Please refer to full WWKS 2+ specification, not required for self-checkout terminals.

8.9 Search Criteria

Please refer to full WWKS 2+ specification, not required for self-checkout terminals.

8.10 Task

Please refer to full WWKS 2+ specification, not required for self-checkout terminals.

8.11 System component (Subscriber)

Description of a participant in a WWKS 2+ communication.

Element	Data type	Description	
Subscriber	Tag	Identification of sender	
Attribute	Data type	Description and values	Default
Id	Integer 32-bit >0	<p>ID of the message sender (here the pharmacy IT system). The ID is used in all further messages to identify their senders and recipients.</p> <p>The ID of the pharmacy IT system is defined in the pharmacy IT system.</p> <p>Possible Values (see 0 Device numbers):</p> <p>100: PIS</p> <p>101..199: PIS device (e.g. cash register)</p> <p>200..999: Device of another vendor</p>	0
Type	String	<p>Type of sending system.</p> <p>Possible values:</p>	"

		<p>"IMS": Pharmacy IT system (PIS),</p> <p>„POS“: Cash register,</p> <p>"Robot": ASRS,</p> <p>„Pickup“: Pick-up terminal,</p> <p>„SelfCheckOut“: Self-Service cash register,</p> <p>„OrderTerminal“: Order terminal,</p> <p>„OTCDisplay“: Digital shelf for OTC products,</p> <p>„SelfServiceDisplay“: Self-service digital shelf,</p> <p>„InformationDisplay“: Information display.</p> <p>This value is for information purpose only and describes the main purpose of the device. For device integration just the list of capabilities is relevant.</p>	
Manufacturer	String	Name of the manufacturer of the sending system	„“
ProductInfo	String	Product name of the sending system	„“
VersionInfo	String	Product version number of the sending system	„“
DeviceName	String	Human readable name of the device that allows to identify it within the pharmacy (e.g. „Cash register 3“)	„“

9 Message reference I – General messages

Mandatory and optional message elements

The messages descriptions contain only the mandatory and typically useful optional elements of the data structures with their special meaning in the context of the respective message. Additional elements can still always be sent with the message. The recipient will process these as far as possible and useful. If an expected optional element is not sent, the recipient should use the default value defined in the data structure.

Example: Batch numbers (`Pack.BatchNumber`) should be included in the pack data structure whenever information about a pack is transmitted, if supported by the PIS/device and known for the pack. The receiver will process it, if supported and useful for the current workflow step. It will assume an empty batch number if it is not sent.

9.1 Message structure

Each message is embedded in the container element `WWKS`, which is constructed according to this scheme:

```
<WWKS Version="2.0" TimeStamp="2013-04-16T11:14:00Z">
</WWKS>
```

The element `WWKS` encloses each individual message. All other elements of a message are sub elements of `WWKS`.

For WWKS 2+, the attribute `Version` has the value "2.0". As WWKS 2+ is backward compatible with WWKS 2, it use the same value as WWKS 2.

The attribute `TimeStamp` is a time stamp in extended UTC format (coordinated universal time). The timestamp is for information purpose only and might be used to determine clock differences when analyzing log files.

In each message, `WWKS` is followed by the "lead" element, which determines the message type.

Sample message:

```
<WWKS Version="2.0" TimeStamp="2018-01-02T11:14:01Z">
  <InputRequest Id="1002" Source="999" Destination="100">
    <Article>
      <Pack ScanCode="01479163"/>
    </Article>
  </InputRequest>
</WWKS>
```

In the example, the lead element `InputRequest` determines the message type: stock input request for a pack. `InputRequest` is followed by additional sub elements further specifying the input request.

9.2 Initialization

Device Support

Device category	Rowa Device	Message support
Self-checkout	Rowa Self-Checkout	Supported

Lead elements

HelloRequest

HelloResponse

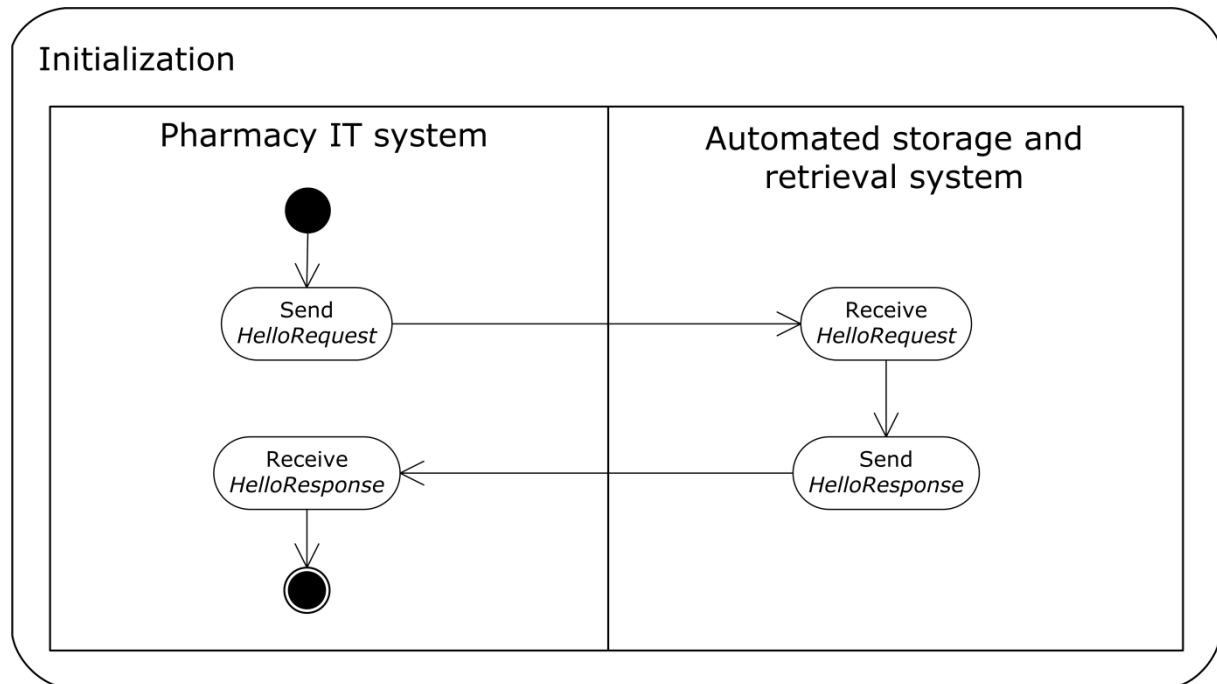
Usage

When the TCP/IP connection between the pharmacy IT system and the automated storage and retrieval system has been established, the client sends the `HelloRequest` once to prepare for the further communications with the communication partner. The server responds with `HelloResponse`. This initializes the connection, readying it for the exchange of further messages.

`HelloRequest` and `HelloResponse` include by a list of supported WWKS capabilities. Both systems thereby indicate which WWKS functions they can process. If there is no listing of supported WWKS capabilities, the receiving system assumes that the sending system supports the capability `"PickingRobot"`.

Optionally, a tenant identifier can be sent in the `HelloRequest`. This identifier is used to distinguish between individual pharmacy IT systems if an automated storage and retrieval system is addressed by several pharmacy IT system at once. All subsequent `Request` messages are then processed in the context of this tenant.

Sequence



9.2.1 HelloRequest

Structure

```

<WWKS>
  <HelloRequest>
    <Subscriber>
      <Capability/>
    </Subscriber>
  </HelloRequest>
</WWKS>

```

Elements

Element	M/O	Data type	Description
HelloRequest	M	Tag	Message type
Attributes	M/O	Data type	Description and Values
Id	M	String	Message ID. This is returned in <i>HelloResponse</i> .

Element	M/O	Data type	Description
Subscriber	M	Tag	Identification of sender
Attributes	M/O	Data type	Description and Values
Id	M	Integer 32-bit >0	<p>ID of the message sender (here the pharmacy IT system). The ID is used in all further messages to identify their senders and recipients.</p> <p>The ID of the pharmacy IT system is defined in the pharmacy IT system.</p> <p>Possible Values (see 5.3 Device numbers):</p> <p>100: PIS</p> <p>101..199: PIS device (e.g. cash register)</p> <p>200..999: Device of another vendor</p>
Type	M	String	<p>Type of sending system.</p> <p>Possible values:</p> <p>"IMS": Pharmacy IT system (PIS),</p> <p>„POS“: Cash register,</p> <p>"Robot": ASRS,</p> <p>„Pickup“: Pick-up terminal,</p> <p>„SelfCheckOut“: Self-Service cash register,</p> <p>„OrderTerminal“: Order terminal,</p> <p>„OTCDisplay“: Digital shelf for OTC products,</p> <p>„SelfServiceDisplay“: Self-service digital shelf,</p> <p>„InformationDisplay“: Information display.</p>
Manufacturer	M	String	Name of the manufacturer of the sending system
ProductInfo	M	String	Product name of the sending system
VersionInfo	M	String	Product version number of the sending system
DeviceName	O	String	Human readable name of the device that allows to identify it within the pharmacy (e.g. „Cash register 3“)

TenantId	O	String	Tenant identifier. Default value is "".
----------	---	--------	---

Element	M/O	Data type	Description
Capability	O	Tag	List of supported WWKS 2 functions. This element can be used multiple times.
Attributes	M/O	Data type	Description and Values
Name	M	String	<p>Name of supported WWKS 2+ - capability. For possible values see chapter 6.</p> <p>Defined values for backwards compability with WWKS2, please refer to WWKS 2 spec for details:</p> <p>"KeepAlive"</p> <p>"Status"</p> <p>"Input"</p> <p>"InitiateInput"</p> <p>"ArticleMaster"</p> <p>"StockDelivery"</p> <p>"StockInfo"</p> <p>"Output"</p> <p>"TaskInfo"</p> <p>"TaskCancel"</p> <p>"Configuration"</p> <p>"StockLocationInfo"</p>

Example

```
<WWKS Version="2.0" TimeStamp="2013-04-16T11:14:00Z">
  <HelloRequest Id="1001">
    <Subscriber Id="333" Type="SelfCheckOut" Manufacturer="MegaDevices"
      ProductInfo="SCO XYZ" VersionInfo="1.4.0">
      <Capability Name="ShoppingCarts"/>
      <Capability Name="BarcodeScan"/>
      <Capability Name="Payment"/>
      <Capability Name="ReceiptLayout"/>
      <Capability Name="ShoppingCartArticleMaster"/>
      <Capability Name="PartialPayment"/>
    </Subscriber>
  </HelloRequest>
</WWKS>
```

Library

TCP/IP connection setup, including initial sending of the `HelloRequest` and receiving of the `HelloResponse` is implemented by the `Connect` method. A mandatory operator is the computer name, or IP address, of the Rowa system. This automatically connects to port 6050. Example:

```
storageSystem.Connect("192.168.64.6");
```

If an alternative port was specified (here 6053), the connection would look like this:

```
storageSystem.Connect("192.168.64.6", 6053);
```

9.2.2 HelloResponse

Structure

```
<WWKS>
  <HelloResponse>
    <Subscriber>
      <Capability/>
    </Subscriber>
  </HelloResponse>
</WWKS>
```

Elements

Element	M/O	Data type	Description
HelloResponse	M	Tag	Message type
Attributes	M/O	Data type	Description and Values
Id	M	String	ID sent in the <i>HelloRequest</i>

Element	M/O	Data type	Description
Subscriber	M	Tag	Identification of sender
Attributes	M/O	Data type	Description and Values
Id	M	Integer 32-bit >0	<p>ID of the message sender (here the pharmacy IT system). The ID is used in all further messages to identify their senders and recipients.</p> <p>The ID of the pharmacy IT system is defined in the pharmacy IT system.</p> <p>Possible Values (see 0</p>

			Device numbers): 100: PIS 101..199: PIS device (e.g. cash register) 200..999: Device of another vendor
Type	M	String	Type of sending system. Possible values: "IMS": Pharmacy IT system (PIS), „POS": Cash register, "Robot": ASRS, picking robot, „Pickup": Pick-up terminal, „SelfCheckOut": Self-Service cash register, „OrderTerminal": Order terminal, „OTCDisplay": Digital shelf for OTC products, „SelfServiceDisplay": Self-service digital shelf, „InformationDisplay": Information display.
Manufacturer	M	String	Name of the manufacturer of the sending system
ProductInfo	M	String	Product name of the sending system
VersionInfo	M	String	Product version number of the sending system
DeviceName	O	String	Human readable name of the device that allows to identify it within the pharmacy (e.g. „Cash register 3")
TenantId	O	String	Tenant identifier. Default value is "".

Element	M/O	Data type	Description
Capability	O	Tag	List of supported WWKS 2 functions. This element can be used multiple times.
Attributes	M/O	Data type	Description and Values

Name	M	String	<p>Name of supported WWKS 2+ - capability. For possible values see chapter 6.</p> <p>Defined values for backwards comp ability with WWKS2, please refer to WWKS 2 spec for details:</p> <p>"KeepAlive"</p> <p>"Status"</p> <p>"Input"</p> <p>"InitiateInput"</p> <p>"ArticleMaster"</p> <p>"StockDelivery"</p> <p>"StockInfo"</p> <p>"Output"</p> <p>"TaskInfo"</p> <p>"TaskCancel"</p> <p>"Configuration"</p> <p>"StockLocationInfo"</p>
------	---	--------	---

Example

```
<WWKS Version="2.0" TimeStamp="2013-04-16T11:14:00Z">
  <HelloResponse Id="1001">
    <Subscriber Id="100" Type="IMS" Manufacturer="SuperPIS" ProductInfo="ThePIS"
VersionInfo="2.0.1">
      <Capability Name="ShoppingCarts"/>
      <Capability Name="BarcodeScan"/>
      <Capability Name="Payment"/>
      <Capability Name="ReceiptLayout"/>
    </Subscriber>
  </HelloResponse>
</WWKS>
```

Library

See *HelloRequest*.

9.3 Keep alive

Device Support

Device category	Rowa Device	Message support
Self-checkout	Rowa Self-Checkout	Supported

Lead elements

KeepAliveRequest

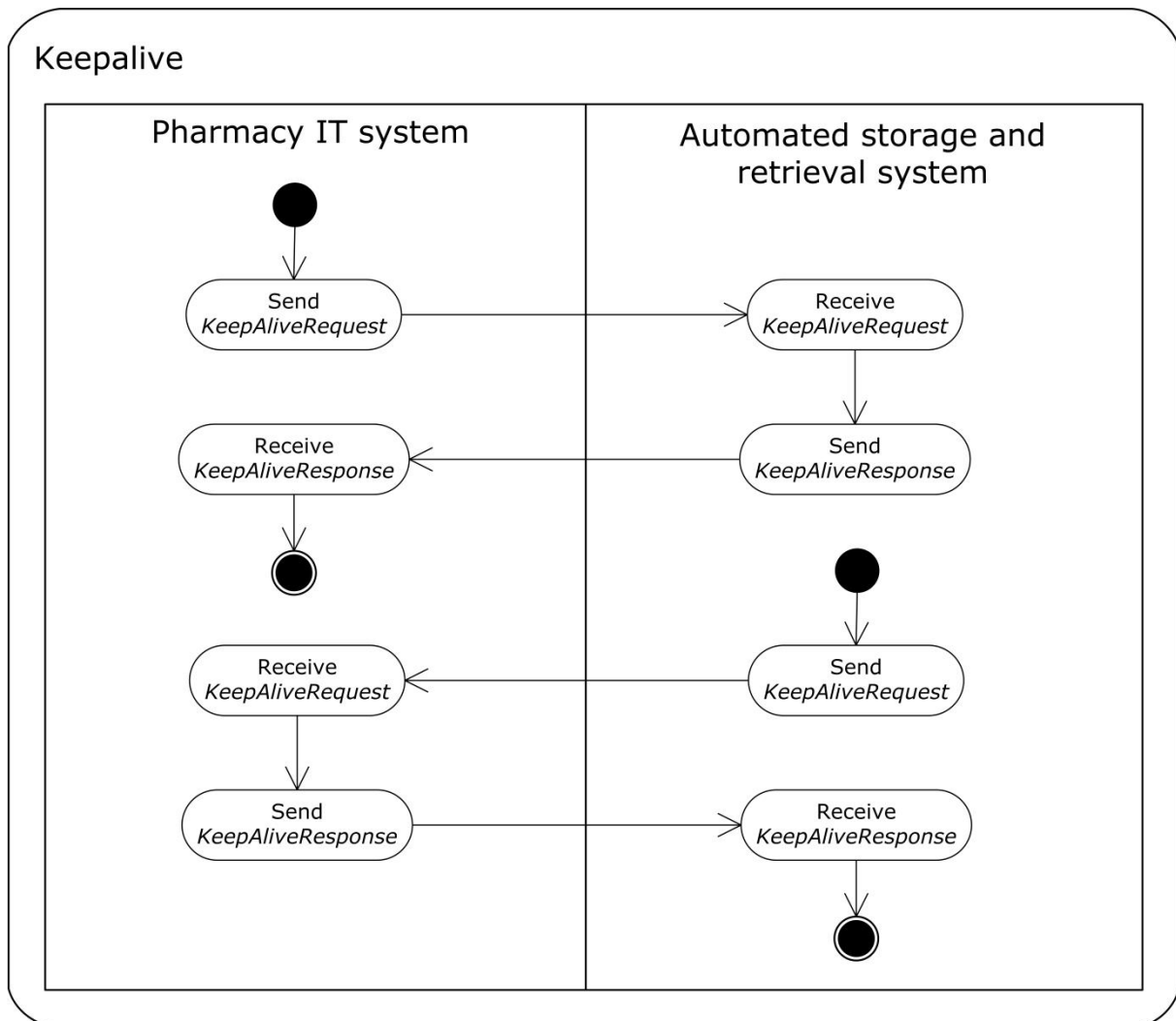
KeepAliveResponse

Usage

A *Keepalive* request can be sent at any time by both systems to check whether the transport channel underlying the connection is still active. This may be useful under the following conditions:

- When the transport channel is potentially unstable (such as with UMTS, GPRS).
- When the network infrastructure being used comprises many active components (e.g. managed switches, routers) and it may be that one of the components has cut the connection.

Sequence



9.3.1 KeepAliveRequest

Structure

```

<WWKS>
  <KeepAliveRequest/>
</WWKS>

```


Elements

Element	M/O	Data type	Description
KeepAliveRequest	M	Tag	Message type
Attributes	M/O	Data type	Description and Values
Id	M	String	Message ID. This is returned in the <i>KeepAliveResponse</i> .
Source	M	Integer 32-bit >0	ID of the system sending the <i>KeepAliveRequest</i>
Destination	M	Integer 32-bit >0	ID of the system intended to receive the <i>KeepAliveRequest</i>

Example

```
<WWKS Version="2.0" TimeStamp="2013-04-16T11:14:00Z">
  <KeepAliveRequest Id="1003" Source="100" Destination="999"/>
</WWKS>
```

Library

The library responds automatically and transparently to every *KeepAliveRequest* from the automated storage and retrieval system.

The library does not currently support a *KeepAliveRequest* from the pharmacy IT system.

9.3.2 KeepAliveResponse**Structure**

```
<WWKS>
  <KeepAliveResponse/>
</WWKS>
```

Elements

Element	M/O	Data type	Description
KeepAliveResponse	M	Tag	Message type
Attributes	M/O	Data type	Description and Values
Id	M	String	Message ID. This is the same as the one sent in the <i>KeepAliveResponse</i> message.

Source	M	Integer 32-bit >0	ID of the system sending the <i>KeepAliveResponse</i>
Destination	M	Integer 32-bit >0	ID of the system intended to receive the <i>KeepAliveResponse</i>

Example

```
<WWKS Version="2.0" TimeStamp="2013-04-16T11:14:00Z">  
  <KeepAliveResponse Id="1003" Source="999" Destination="100"/>  
</WWKS>
```

Library

The library responds automatically and transparently to every *KeepAliveRequest* from the automated storage and retrieval system.

The library does not currently support a *KeepAliveRequest* from the pharmacy IT system.

9.4 Unknown messages

Device Support

Device category	Rowa Device	Message support
Self-checkout	Rowa Self-Checkout	<i>Under development</i>

Lead elements

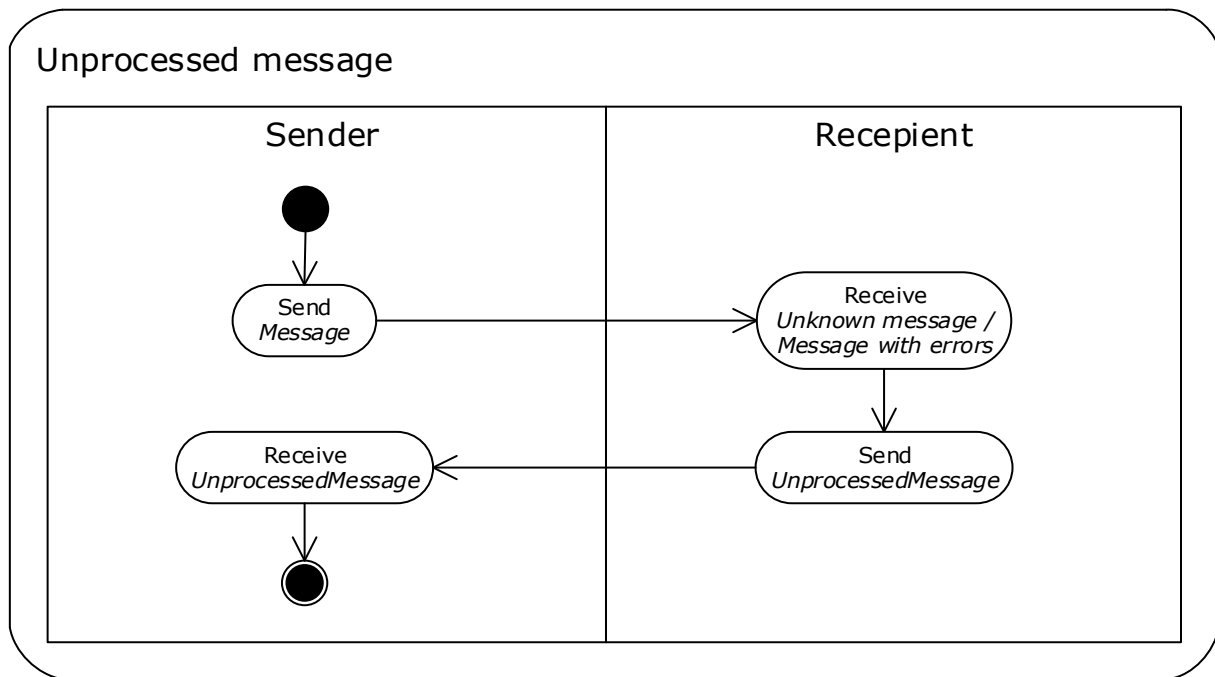
UnprocessedMessage

Usage

If PIS or device receive a message, they cannot understand or process, they may tell the other side about this fact by sending an *UnprocessedMessage*.

The receiver of this message should not process it automatically except writing it into a log file. This log might help analyzing issues when developing the interface integration.

Sequence



9.4.1 UnprocessedMessage

Structure

```

<WWKS>
  <UnprocessedMessage/>
</WWKS>

```

Elements

Element	M/O	Data type	Description
UnprocessedMessage	M	Tag	Message type
Attributes	M/O	Data type	Description and values
Id	M	String64	ID of the message.
Source	M	Integer 32-bit >0	ID of the system component that sends the message.
Destination	P	Integer 32-bit >0	ID of the message destination system component

Reason	O	String	Reason of non-processing the original message. Possible values: "SyntaxError": Invalid XML or other syntactical errors "NotSupported": Receiver does not support this message type. "DataError": Receiver expected additional attributes in the message or received unknown data it was not able to handle. Used only if no specific Response is defined to handle this error.
Text	O	String	Human readable text with additional hints to reason.

Element	M/O	Data type	Description
Message	M	Tag	Information of received but unprocessed message. The message content is embedded as CDATA XML into this element.
Id	O	String64	ID of received message, if detected.

Example

```
<WWKS Version="2.0" Timestamp="2018-03-14T11:14:01Z">
  <UnprocessedMessage Id="3335" Source="100" Destination="999"
    Reason="SyntaxError" Text="Missing WWKS tag">
    <Message>
      <![CDATA[
        <WWX Version="2.0" Timestamp="2018-03-14T11:14:00Z">
          <InputRequest Id="1002" Source="999" Destination="100">
            <Article>
              <Pack ScanCode="12156894"/>
            </Article>
          </InputRequest>
        </WWX>
      ]]>
    </Message>
```

```
</UnprocessedMessage>
</WWKS>

<WWKS Version="2.0" TimeStamp="2018-03-14T12:14:01Z">
  <UnprocessedMessage Id="3336" Source="100" Destination="999"
    Reason="NotSupported" Text="Unknown Message ArticlePriceRequest">
    <Message Id="1101">
      <![CDATA[
        <WWKS Version="2.0" TimeStamp="2018-03-14T12:14:00Z">
          <ArticlePriceRequest Id="1101" Source="100" Destination="999"
            Currency="EUR">
            <Article Id="Article-1234"/>
          </ArticlePriceRequest>
        </WWKS>
      ]]>
    </Message>
  </UnprocessedMessage>
</WWKS>
```

Library

The library sends an `UnprocessedMessage` in case it receive a message it cannot process.

It logs all received `UnprocessedMessage`.

10 Message reference II – article data

10.1 Master data

Please refer to full WWKS2+ specification, not required for pick-up terminals.

10.2 Good receipt

Please refer to full WWKS2+ specification, not required for pick-up terminals.

10.3 Article Information

Deprecated for use with self-checkout terminals. All required article data are now available in the ShoppingCart. Please use ShoppingCart messages instead.

For message description please refer to full WWKS2+ specification.

10.4 Serial number validation

Please refer to full WWKS2+ specification, not required for self-checkout terminals.

11 Message reference III – Automated storage and retrieval system

Please refer to full WWKS2+ specification, not required for self-checkout terminals.

12 Message reference IV – Sales transactions

12.1 Request shopping cart/sales transaction

Device support

Device category	Rowa Device	Message support
Self-checkout	Rowa Self-Checkout	Supported

Lead elements

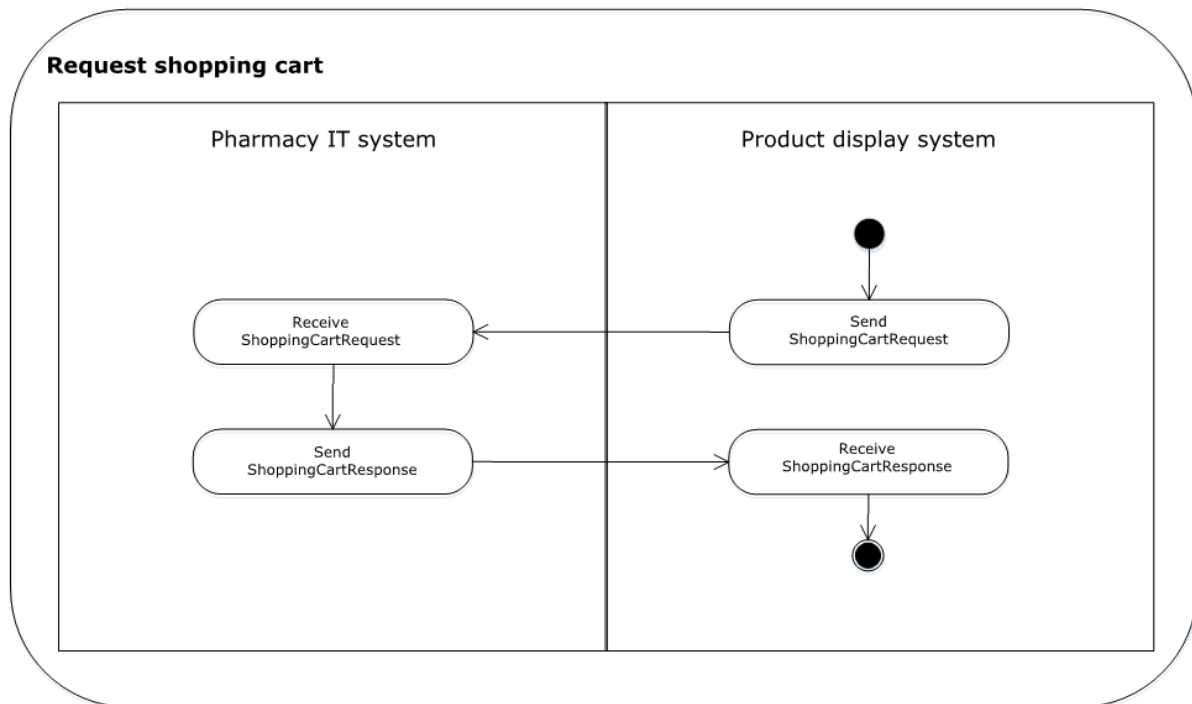
ShoppingCartRequest

ShoppingCartResponse

Usage

`ShoppingCartRequest` can be used by the behind-the-counter system to request both an existing and a new shopping cart/sales transaction. The pharmacy IT system responds with a `ShoppingCartResponse` containing the corresponding shopping cart/sales transaction.

Sequence



12.1.1 ShoppingCartRequest

Device support

Device category	Rowa Device	Message support
Self-checkout	Rowa Self-Checkout	Supported

Structure

```

<WWKS>
  <ShoppingCartRequest>
    <Criteria/>
  </ShoppingCartRequest>
</WWKS>

```

Elements

Element	M/O	Data type	Description
ShoppingCartRequest	M	Tag	Message type
Attributes	M/O	Data type	Description and Values

Id	M	String	ID of the shopping cart request process.
Source	M	Integer 32-bit >0	ID of the system sending the <i>ShoppingCartRequest</i> message.
Destination	M	Integer 32-bit >0	ID of the system intended to receive the <i>ShoppingCartRequest</i> message.

Element	M/O	Data type	Description
Criteria	M	Tag	Shopping cart filters follow. This element may occur once only.
Attributes	M/O	Data type	Description and Values
ShoppingCartId	O	String64	ID of a shopping cart/order process.
SalesPointId	O	String64	ID of a point of sale.
<i>ViewPointId</i>	O	String	<i>Deprecated. Devices have no longer to send this id.</i> ID of a behind-the-counter system.
SalesPersonId	O	String64	ID of a salesperson.
CustomerId	O	String64	ID of a customer.
PickupId	O	String64	Pick-up code shared with the patient.
DisplayLanguage	O	String	ISO-639-1 code of the current display language of the device.

Example

```
<WWKS Version="2.0" TimeStamp="2013-04-16T11:14:00Z">
  <ShoppingCartRequest Id="1103" Source="100" Destination="999">
    <Criteria PickupId="PU64738"/>
  </ShoppingCartRequest>
</WWKS>
```


Library

```

var digitalShelf = new RowaDigitalShelf();
digitalShelf.ShoppingCartRequested += DigitalShelf_ShoppingCartRequested;

void DigitalShelf_ShoppingCartRequested(IDigitalShelf sender,
    IShoppingCartRequest request)
    {
        Console.WriteLine(string.Format("Shopping cart has been requested
from the digital shelf.));
        var shoppingCart =
digitalShelf.CreateShoppingCart(request.Criteria.ShoppingCartId,
ShoppingCartStatus.Active, "CustID", "SaleID", "salePointId", "ViewPointID");
        shoppingCart.AddItem("1", 10, 2, 5, "100", "EUR");
        shoppingCart.AddItem("2", 50, 10, 50, "10", "FR");
        request.Accept(shoppingCart);
        shoppingCart.Status = ShoppingCartStatus.Active;
        shoppingCart.SalesPointId = "1";
        shoppingCart.ViewPointId = "2";
        shoppingCart.SalesPersonId = "3";
        shoppingCart.CustomerId = "4";
        digitalShelf.UpdateShoppingCart(shoppingCart);
    }

```

12.1.2 ShoppingCartResponse**Device support**

Device category	Rowa Device	Message support
Self-checkout	Rowa Self-Checkout	Supported

Structure

```

<WWKS>
  <ShoppingCartResponse>
    <ShoppingCart>
      <ShoppingCartItem>
      </ShoppingCartItem>
    </ShoppingCart>
  </ShoppingCartResponse>
</WWKS>

```

Elements

Element	M/O	Data type	Description
---------	-----	-----------	-------------

ShoppingCartResponse	M	Tag	Message type
Attributes	M/O	Data type	Description and Values
Id	M	String64	ID of the shopping cart request process.
Source	M	Integer 32-bit >0	ID of the system sending the <i>ShoppingCartResponse</i> message.
Destination	M	Integer 32-bit >0	ID of the system intended to receive the <i>ShoppingCartResponse</i> message.
Description	O	String	Human readable text with information about the transaction or reason for reject. To be displayed on the device.

Element	M/O	Data type	Description
ShoppingCart	O	Tag	The shopping cart/order process provided by the pharmacy IT system. This element may occur multiple.

Example

```
<WWKS Version="2.0" TimeStamp="2013-04-16T11:14:00Z">
  <ShoppingCartResponse Id="1103" Source="999" Destination="100">
    <ShoppingCart Id="SC-2345" SalesPointId="103 Status="Active" PickupId="PU64738"
      SalesPersonId="SalesPerson-4711" CustomerId="Customer-42">
      <ShoppingCartItem ArticleId=" 1111" Name="Article-1111"
        OrderedQuantity="1" DispensedQuantity="0"
        PaidQuantity="1" Price"4.95" Currency="EUR" VAT="19.0"/>
    </ShoppingCart>
  </ShoppingCartResponse>
</WWKS>
```

Library

See ShoppingCartRequest.

12.2 Modify shopping cart/sales transaction

Device support

Device category	Rowa Device	Message support
Self-checkout	Rowa Self-Checkout	Supported, update elements under development

Lead elements

ShoppingCartUpdateRequest

ShoppingCartUpdateResponse

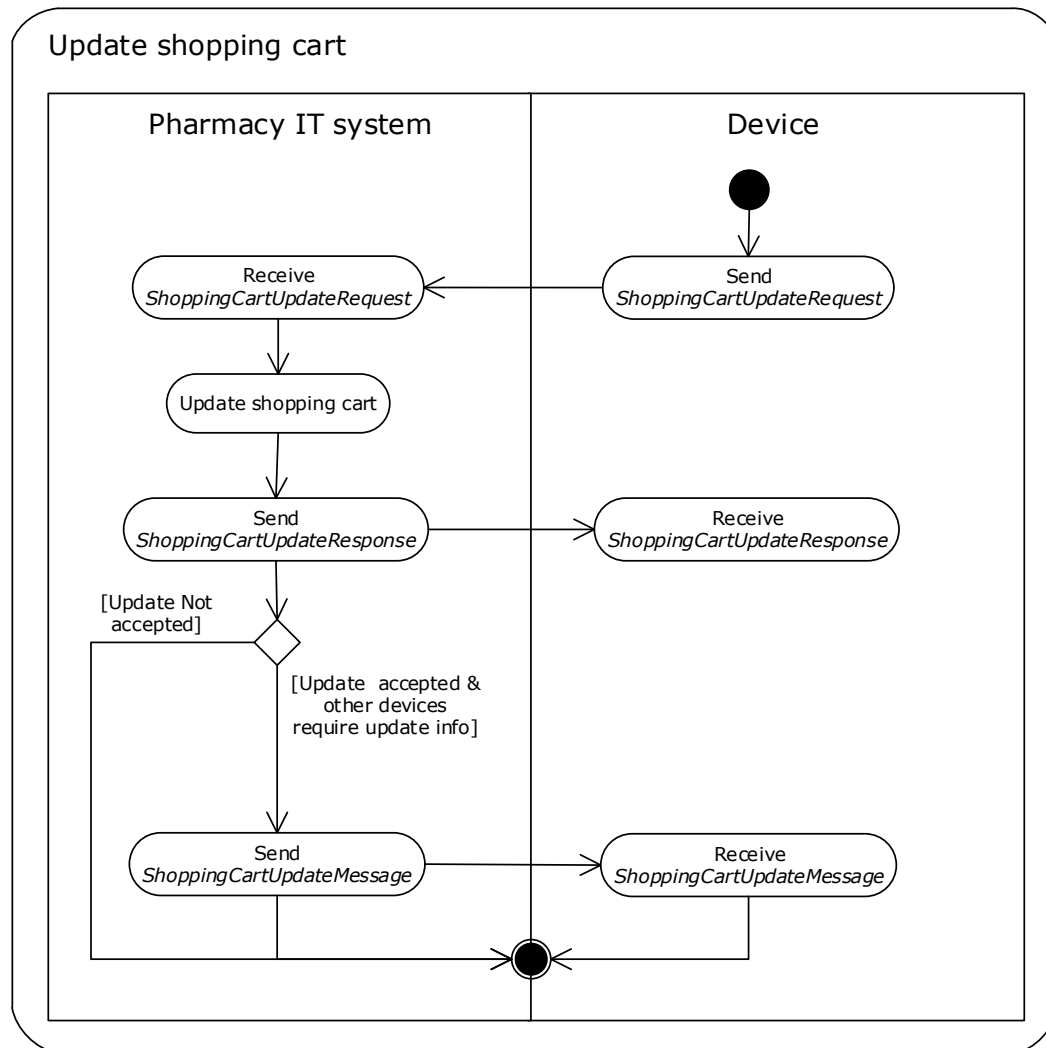
Usage

ShoppingCartUpdateRequest is used by a behind-the-counter system to request changes to a shopping cart/order process. Changes include discarding or concluding an order process, adding and deleting items or changing the number of units for an item. The complete shopping cart is transferred in all cases.

The pharmacy IT system responds to the request with a *ShoppingCartUpdateResponse*, which provides information on the status of the change and, if other devices are attached to the same sales point concludes by publishing the current shopping cart in a *ShoppingCartUpdateMessage*.

Support of the *ShoppingCartUpdateMessage* is not required for Self-Checkout or Pickup terminals.

Sequence



12.2.1 ShoppingCartUpdateRequest

Device support

Device category	Rowa Device	Message support
Self-checkout	Rowa Self-Checkout	Supported update elements under development

Structure

```

<WWKS>
  <ShoppingCartUpdateRequest>
    <BarcodeScanned/>
    <Add/>
    <Delivery/>
  
```

```

<ChangeQuantity/>
<Payment>

    <PaymentItem/>
</Payment>
<PaymentRequest>
    <PaymentItem/>
</PaymentRequest>
<Transfer/>
<Language/>
<CartUpdate/>
<QueryAnswer/>
<Close/>
<ShoppingCart>
    <ShoppingCartItem/>
    <PaymentItem/>
    <Receipt/>
</ShoppingCart>
</WWKS>

```

Elements

Element	M/O	Data type	Description
ShoppingCartUpdateRequest	M	Tag	Message type
Attributes	M/O	Data type	Description and Values
Id	M	String64	ID of the shopping cart change process.
Source	M	Integer 32-bit >0	ID of the system sending the <i>ShoppingCartUpdateRequest</i> message.
Destination	M	Integer 32-bit >0	ID of the system intended to receive the <i>ShoppingCartUpdateRequest</i> message.
ShoppingCartId	M	String64	ID of shopping cart the device want to request an update for. <i>In previous versions of WWKS2+ this id was only provided in the ShoppingCart element which is now optional and for error analysis purpose only.</i>
Action (Deprecated)	O	String	<i>Deprecated, use update request element instead.</i> Type of modification.

			<p>Each update request should contain only similar changes . The PIS may ignore modification not matching the action value.</p> <p>Possible values:</p> <p>„Order“: Add or remove shopping cart lines. Change the order quantity of shopping cart lines. To add a lines, the device has to know the article id of the item, it must not use the barcode scanned.</p> <p>„Pricing“: Change of the price of the shopping cart or one or multiple shopping cart lines.</p> <p>„Delivery“: Change the delivered quantity.</p> <p>„Payment“: Change of paid quantity.</p> <p>„PaymentRequest“: Request to the PIS to handle the payment of the shopping cart and to return the transaction details in the response.</p> <p>„Transfer“: Modify point of sale, customer, sales person or other generic attributes of the sales transaction.</p> <p>„BarcodeScan“: A barcode was scanned that might be relevant for the shopping cart. The content of the barcode is transferred in the <code>Barcode</code> attribute.</p> <p>„LoyaltyCardScanned“: A loyalty card or another card identifying the customer has been scanned. The content of the barcode is transferred in the <code>Barcode</code> attribute.</p> <p>„CouponScanned“: A coupon has been scanned. The PIS should apply the discount defined by the coupon to the shopping cart. The content of the barcode is transferred in the <code>Barcode</code> attribute.</p> <p>„CartUpdate“: Request to take over full shopping cart as provided by device in the request.</p> <p>Default value: „CartUpdate“.</p>
Barcode (Deprecated)	O	String	<p><i>Deprecated. Use <code>BarcodeScanned</code> update request element instead.</i></p> <p>Barcode scanned. E.g. Item code, loyalty card or coupon. Used in combination with</p>

			Action attribute values BarcodeScan, LoyaltyCard, Coupon
--	--	--	---

At least one of the update request elements listed below should appear in the request. If no update request element is given, the IT system assumes that a <CartUpdate> element is used:

Element	M/O	Data type	Description
BarcodeScanned	O	Tag	Request to modify a shopping cart according to a barcode scanned at the device by a user. May appear only once.
Attributes	M/O	Data type	Description and Values
Barcode	M	String	ID of the article to be added.
Type	O	String	Type of the scanned barcode. Possible values: "Unknown": Device does not know what kind of barcode has been scanned by the user. "Article": Barcode of an article. The barcode might be different to the article id. "LoyaltyCard": Barcode of a loyalty card. "Coupon": Barcode of a discount coupon. Default value: "Unknown"

Element	M/O	Data type	Description
Add	O	Tag	Request to add a line to the shopping cart. May appear only once.
Attributes	M/O	Data type	Description and Values
ArticleId	M	String64	ID of the article to be added.
OrderedQuantity	M	Int	Ordered quantity of the article.

Element	M/O	Data type	Description
Delivery	O	Tag	Request to the IT system to accept delivery all or parts of the ordered quantity of an article in the shopping cart. May appear multiple.
Attributes	M/O	Data type	Description and Values
ArticleId	M	String64	ID of the article to be added.
DispensedQuantity	M	Int	Number of packs delivered in this action.

Element	M/O	Data type	Description
ChangeQuantity	O	Tag	Request to modify the ordered quantity of an article in the shopping cart. May appear multiple.
Attributes	M/O	Data type	Description and Values
ArticleId	M	String64	ID of the article to be added.
OrderedQuantity	M	Int	New (absolute) number of packs ordered for the article.
LastOrderedQuantity	O	Int	Number of packs ordered for the pack before the change of quantity now requested (as far as the device knows).

Element	M/O	Data type	Description
PaymentRequest	O	Tag	Request to the IT system to perform payment of the shopping cart. May appear only once.
Attributes	M/O	Data type	Description and Values
ReceiptWidth	O	Integer 32-Bit > 0	Number of characters the receipt printer of the device is able to print in one row. A value of 0 means undefined.

			Default value is 0.
--	--	--	---------------------

Element	M/O	Data type	Description
Payment	O	Tag	Request to the IT system to accept full or partial payment of the shopping cart. May appear only once.
ReceiptWidth	O	Integer 32-Bit > 0	Number of characters the receipt printer of the device is able to print in one row. A value of 0 means undefined. Default value is 0.

Element	M/O	Data type	Description
PaymentItem	M	Tag	Detail information of the Payment. May appear only once.

Element	M/O	Data type	Description
Language	O	Tag	Request to modify the display language of the shopping cart. May appear only once.
Attributes	M/O	Data type	Description and Values
DisplayLanguage	M	String	ISO-639-1 code of new display language.

Element	M/O	Data type	Description
CartUpdate	O	Tag	Request to replace current shopping cart with the content of the ShoppingCart in this ShoppingCartUpdateRequest. Defined to maintain backwards compatibility with older version of WWKS2+ that did not define update request elements.

Attributes	M/O	Data type	Description and Values
<i>none</i>			

Element	M/O	Data type	Description
Transfer	O	Tag	<p><i>Not used by self-checkout terminals</i></p> <p>Request to transfer the shopping cart to another sales point.</p> <p>May appear only once. Must not be combined with other update request elements</p>
Attributes	M/O	Data type	Description and Values
SalesPointId	M	String64	<p>ID of the sales point to transfer the shopping cart to.</p> <p>Empty to indicate that devices has finished handling the shopping cart and wants to return it to the IT system.</p> <p>Default value is "".</p>

Element	M/O	Data type	Description
Close	O	Tag	<p>Request to finish handling this shopping cart at the device.</p> <p>May appear only once. Must not be combined with other update request elements</p>
Attributes	M/O	Data type	Description and Values
<i>none</i>			

Element	M/O	Data type	Description
QueryAnswer	O	Tag	Answer provided by the user of the device in response to a ShoppingCartUpdateResponse with UpdateResult.Status="Query". May appear only once.
Attributes	M/O	Data type	Description and Values
Id	M	String64	Id of the query answer option that has been selected by the user of the device.

Element	M/O	Data type	Description
ShoppingCart	O	Tag	Details of the sales updated transaction from the device point of. The IT system usually ignores this and returns the modified shopping cart according to its own state. This element may occur once only.

Example

```
<WWKS Version="2.0" TimeStamp="2013-04-16T11:14:00Z">
  <ShoppingCartUpdateRequest Id="1104" Source="100" Destination="999"
    ShoppingCartId="SC-2345">
    <BarcodeScanned Type="Unknown" Barcode="857673463"/>
  </ShoppingCartUpdateRequest>
</WWKS>
```

Library

The library currently just supports the deprecated usage of ShoppingCartUpdateRequest without action attribute and update elements.

```
var digitalShelf = new RowaDigitalShelf();
digitalShelf.ShoppingCartUpdateRequested +=
DigitalShelf_ShoppingCartUpdateRequested;
void DigitalShelf_ShoppingCartUpdateRequested(IDigitalShelf sender,
IShoppingCartUpdateRequest request)
{
    Console.WriteLine(string.Format("Shopping cart update has been
requested from the digital shelf."));
    // process update
    request.Accept("Shopping cart has been updated successfully");
}
```

12.2.2 ShoppingCartUpdateResponse

Device support

Device category	Rowa Device	Message support
Self-checkout	Rowa Self-Checkout	Supported

Structure

```
<WWKS>
  <ShoppingCartUpdateResponse>
    <ShoppingCart/>
    <UpdateResult>
      <Query/>
    </UpdateResult/>
  </ShoppingCartUpdateResponse>
</WWKS>
```

Elements

Element	M/O	Data type	Description
ShoppingCartUpdateResponse	M	Tag	Message type
Attributes	M/O	Data type	Description and Values
Id	M	String64	ID of the shopping cart change process.
Source	M	Integer 32-bit >0	ID of the system sending the <i>ShoppingCartUpdateResponse</i> message.
Destination	M	Integer 32-bit >0	ID of the system intended to receive the <i>ShoppingCartUpdateResponse</i> message.
Action	O	String	<i>Deprecated.</i> Type of modification as defined in the <i>ShoppingCartUpdateRequest</i> .

Element	M/O	Data type	Description
ShoppingCart	M	Tag	Details of the sales transaction. This element may occur once only.

Element	M/O	Data type	Description
UpdateResult	M	Tag	Details the status of the shopping cart update. This element may occur once only.
Status	M	String	The type of shopping cart change requested. Possible values: "Updated", if the requested update has been performed. "NotUpdated" if the requested update has not been performed. "Query", if further information is required from the user. Possible answer options are provided in the <code>Query</code> elements.
Description	O	String	Free text detailing additional information on the status of the update. The text is normally used when rejecting an update to provide detailed information on the reasons for rejecting the update or for querying additional information from the user. The text may be displayed on the behind-the-counter system, and should therefore be localized according to the <code>DisplayLanguage</code> of the shopping cart.

Element	M/O	Data type	Description
Query	M/O	Tag	Mandatory in case Status="Query". Provides an answer option that the device presents to the user. May appear multiple.
Id	M	String64	ID of the answer option.
Text	M	String	Text of the answer option.

Example

```

<WWKS Version="2.0" TimeStamp="2013-04-16T11:14:00Z">
  <ShoppingCartUpdateResponse Id="1104" Source="999" Destination="100">
    <ShoppingCart Id="SC-2345">
      <ShoppingCartItem ArticleId="1111" Name="Article-1111" OrderedQuantity="5"/>
      <ShoppingCartItem ArticleId="1111" Name="Article-1111" OrderedQuantity="1"
        DispensedQuantity="0" PaidQuantity="0" Price="15.00" VAT="19.0">
        <Article Id="1111">
          <PriceInformation Category="Standard" Description="List price"
            Quantity="1" Price="20.00" BasePrice="10.00" BasePriceUnit="100ml"
            VAT="19"/>
          <PriceInformation Category="Offer" Description="Offer" Quantity="1"
            Price="15.00" BasePrice="7.50" BasePriceUnit="100ml" VAT="19"/>
          <Tag Value="AgeVerification"/>
        </Article>
      </ShoppingCartItem>
      <ShoppingCartItem ArticleId="2222" Name="Article-2222" OrderedQuantity="4"
        DispensedQuantity="4" PaidQuantity="4" Price="18.95" VAT="19.0"/>
      <Article Id="2222">
        <PriceInformation Category="Standard" Description="List price"
          Quantity="1" Price="18.95" BasePrice="18.95" BasePriceUnit="10"
          VAT="19"/>
        <Tag Value="Consultation"/>
      </Article>
    </ShoppingCartItem>
  </ShoppingCart>
  <UpdateResult Status="Updated" Description="Item has been added."/>
</ShoppingCartUpdateResponse>
</WWKS>

```

Library

See *ShoppingCartUpdateRequest*.

12.2.3 ShoppingCartUpdateMessage

Please refer to full WWKS2+ specification, not required for self-checkout terminals.

12.3 Article selection

Please refer to full WWKS2+ specification, not required for self-checkout terminals.