XFS4IoT Preview Messaging for Dispenser Draft 0.0.3 documentation

Introduction

Documentation

Commands

Unsolicited Events

- Dispenser.MediaDetectedEvent
- Dispenser.ShutterStatusChangedEvent

Events

- CashManagement.CashUnitErrorEvent
- CashManagement.CashUnitThresholdEvent
- Dispenser.DelayedDispenseEvent
- Dispenser.StartDispenseEvent
- Dispenser.PartialDispenseEvent
- Dispenser.SubDispenseOkEvent
- Dispenser.IncompleteDispenseEvent
- CashManagement.NoteErrorEvent
- CashManagement.InputP6Event
- CashManagement.InfoAvailableEvent
- Dispenser.IncompleteRetractEvent
- CashManagement.CashUnitInfoChangedEvent

XFS4IoT Preview Messaging for Dispenser Draft 0.0.3

This specification describes the functionality of an XFS4IoT compliant Cash Dispenser interface. It defines the service-specific commands that can be issued to the service using the WebSocket endpoint.

Persistent values are maintained through power failures, open sessions, close session and system resets.

This specification covers the dispensing of items. An "item" is defined as any media that can be dispensed and includes coupons, documents, bills and coins.

All currency parameters in this specification are expressed as a quantity of minimum dispense units, as defined in the description of the CashManagement.CurrencyExp command.

Commands

Dispenser.Status

Description

This command is used to obtain the status of the Dispenser. It may also return vendor-specific status information.

Command Message

Message Header

Name	Type	Default	Description
requestId (Required)	string		Unique request identifier supplied by the client used to correlate the command with responses, events and completions. For Unsolicited Events the field will be empty.
type (Required)	string		The message type, either command, response, event or completion.
name (Required)	string		The original message name, for example "CardReader.Status"

Message Payload

Name	Type	Default	Description
timeout	integer	0	Timeout in milliseconds for the command to complete. If set to zero, the command will not timeout but can be cancelled.

Example Message (generated)

```
{
  "headers": {
    "requestId": "b34800d0-9dd2-4d50-89ea-92d1b13df54b",
    "type": "command",
    "name": "string"
  },
  "payload": {
    "timeout": "5000"
  }
}
```

Completion Message

Message Header

Name	Type	Default	Description
requestId (Required)	string		Unique request identifier supplied by the client used to correlate the command with responses, events and completions. For Unsolicited Events the field will be empty.
type (Required)	string		The message type, either command, response, event or completion.
name (Required)	string		The original message name, for example "CardReader.Status"

Message Payload

Name	Type	Default	Description
status	string		ok if the commmand was successful otherwise error
errorDescription	string		If error, identified that cause of the error
device	string		Specifies the state of the device.
extra	array		Specifies a list of vendor-specific, or any other extended, information. The information is returned as a series of "key=value" strings so that it is easily extendable by Service Providers.
guidLights	array		Specifies the state of the guidance light indicators. A number of guidance light types are defined below. Vendor specific guidance lights are defined starting from the end of the array.

Name	Type	Default	Description
guidLights.flashRate	string		Indicates the current flash rate of the guidelight.
guidLights.color	string		Indicates the current color of the guidelight.
guidLights.direction	string		Indicates the current direction of the guidelght.
devicePosition	string		Position of the device.
powerSaveRecoveryTime	integer		Specifies the actual number of seconds required by the device to resume its normal operational state from the current power saving mode. This value is zero if either the power saving mode has not been activated or no power save control is supported
antiFraudModule	string		Specifies the state of the anti-fraud module
	object		
safeDoor	string		Supplies the state of the safe door. Following values are possible: "doorNotSupported": Physical device has no safe door or safe door state reporting is not supported. "doorOpen": Safe door is open. "doorClosed": Safe door is closed.'

Name	Type	Default	Description
dispenser	string		Supplies the state of the dispenser’s logical cash units. Following values are possible: "ok": All cash units present are in a good state. "cashUnitState": One or more of the cash units is in a low, empty, inoperative or manipulated condition. Items can still be dispensed from at least one of the cash units. "cashUnitStop": Due to a cash unit failure dispensing is impossible. No items can be dispensed because all of the cash units are in an empty, inoperative or manipulated condition. This state may also occur when a reject/retract cash unit is full or no reject/retract cash unit is present, or when an application lock is set on every cash unit which can be locked. "cashUnitUnknown": Due to a hardware error or other condition, the state of the cash units cannot be determined.

Name	Type	Default	Description
intermediateStacker	string		Supplies the state of the intermediate stacker. These bills are typically present on the intermediate stacker as a result of a retract operation or because a dispense has been performed without a subsequent present. Following values are possible: "empty": The intermediate stacker is empty. "notEmpty": The intermediate stacker is not empty. The items have not been in customer access. "notEmptyCustomer": The intermediate stacker is not empty. The items have been in customer access. If the device is a recycler then the items on the intermediate stacker may be there as a result of a previous cash-in operation. "notEmptyUnknown": The intermediate stacker is not empty. It is not known if the items have been in customer access. "unknown": Due to a hardware error or other condition, the state of the intermediate stacker cannot be determined. "notSupported": The physical device has no intermediate stacker.
positions	array		Array of structures for each position to which items can be dispensed or presented.

Name	Type	Default	Description
positions.position	string		Supplies the output position as one of the following values: "left": Left output position. "right": Right output position. "center": Center output position. "top": Top output position. "bottom": Bottom output position. "front": Front output position. "rear": Rear output position.
positions.shutter	string		Supplies the state of the shutter. Following values are possible: "closed": The shutter is operational and is closed. "open": The shutter is operational and is open. "jammed": The shutter is jammed and is not operational. The field jammedShutterPosition provides the positional state of the shutter. "unknown": Due to a hardware error or other condition, the state of the shutter cannot be determined. "notSupported": The physical device has no shutter or shutter state reporting is not supported.

Name	Type	Default	Description
positions.positionStatus	integer		Returns information regarding items which may be at the output position. If the device is a recycler it is possible that the output position will not be empty due to a previous cash-in operation. Following values are possible: "empty": The output position is empty. "notEmpty": The output position is not empty. "unknown": Due to a hardware error or other condition, the state of the output position cannot be determined. "notSupported": The device is not capable of reporting whether or not items are at the output position.
positions.transport	string		Supplies the state of the transport mechanism. The transport is defined as any area leading to or from the position. Following values are possible: "ok": The transport is in a good state. "inoperative": The transport is inoperative due to a hardware failure or media jam. "unknown":Due to a hardware error or other condition the state of the transport cannot be determined. "notSupported": The physical device has no transport or transport state reporting is not supported.

Name	Type	Default	Description
positions.transportStatus	string		Returns information regarding items which may be on the transport. If the device is a recycler device it is possible that the transport will not be empty due to a previous cash-in operation. Following values are possible: "empty": The transport is empty. "notEmpty": The transport is not empty. "notEmptyCustomer": Items which a customer has had access to are on the transport. "notEmptyUnkown": Due to a hardware error or other condition it is not known whether there are items on the transport. "notSupported": The device is not capable of reporting whether items are on the transport.
positions.jammedShutterPosition	string		Returns information regarding the position of the jammed shutter. Following values are possible: "notSupported": The physical device has no shutter or the reporting of the position of a jammed shutter is not supported. "notJammed": The shutter is not jammed. "open": The shutter is jammed, but fully open. "partiallyOpen": The shutter is jammed, but partially open. "closed": The shutter is jammed, but fully closed. "unknown": The position of the shutter is unknown.

Name	Type	Default	Description

Example Message (generated)

```
{
  "headers": {
    "requestId": "b34800d0-9dd2-4d50-89ea-92d1b13df54b",
    "type": "command",
    "name": "string"
  },
  "payload": {
    "status": "ok",
    "errorDescription": "string",
    "device": "online",
    "extra": [
      "string"
    ],
    "guidLights": [
      {}
    ],
    "devicePosition": "inposition",
    "powerSaveRecoveryTime": 0,
    "antiFraudModule": "notSupp",
    "safeDoor": "doorNotSupported",
    "dispenser": "ok",
    "intermediateStacker": "empty",
    "positions": [
      {
        "position": "left",
        "shutter": "closed",
        "positionStatus": 0,
        "transport": "ok",
        "transportStatus": "empty",
        "jammedShutterPosition": "notSupported"
      }
    ]
  }
}
```

Event Messages

Dispenser.Capabilities

Description

This command retrieves the capabilities of the Dispenser. It may also return vendor specific capability information. The intermediate stacker and the transport are treated as separate areas. Some devices may have the capability to move items from the cash units to the intermediate stacker while there are items on the transport. Similarly some devices may be able to retract items to the transport or the cash units while there are items on the intermediate stacker.

Command Message

Message Header

Name	Type	Default	Description

Name	Type	Default	Description
requestId (Required)	string		Unique request identifier supplied by the client used to correlate the command with responses, events and completions. For Unsolicited Events the field will be empty.
type (Required)	string		The message type, either command, response, event or completion.
name (Required)	string		The original message name, for example "CardReader.Status"

Message Payload

Name	Type	Default	Description
timeout	integer	0	Timeout in milliseconds for the command to complete. If set to zero, the command will not timeout but can be cancelled.

Example Message (generated)

```
{
  "headers": {
    "requestId": "b34800d0-9dd2-4d50-89ea-92d1b13df54b",
    "type": "command",
    "name": "string"
  },
  "payload": {
    "timeout": "5000"
  }
}
```

Completion Message

Message Header

Name	Type	Default	Description
requestId (Required)	string		Unique request identifier supplied by the client used to correlate the command with responses, events and completions. For Unsolicited Events the field will be empty.
type (Required)	string		The message type, either command, response, event or completion.
name (Required)	string		The original message name, for example "CardReader.Status"

Message Payload

Name	Type	Default	Description
status	string		ok if the commmand was successful otherwise error
errorDescription	string		If error, identified that cause of the error

Name	Type	Default	Description
class	string		Specifies the logical service class
compound	boolean		Specifies whether the logical device is part of a compound physical device
extra	array		Specifies a list of vendor-specific, or any other extended, information. The information is returned as a series of "key=value" strings so that it is easily extendable by Service Providers
guidLights	array		Specifies which guidance lights are available
guidLights.flashRate	object		Indicates which flash rates are supported by the guidelight.
guidLights.flashRate.slow	boolean		The light can blink slowly.
guidLights.flashRate.medium	boolean		The light can blink medium frequency.
guidLights.flashRate.quick	boolean		The light can blink quickly.
guidLights.flashRate.continuous	boolean		The light can be continuous (steady).
guidLights.color	object		Indicates which colors are supported by the guidelight.
guidLights.color.red	boolean		The light can be red.
guidLights.color.green	boolean		The light can be green.
guidLights.color.yellow	boolean		The light can be yellow.
guidLights.color.blue	boolean		The light can be blue.
guidLights.color.cyan	boolean		The light can be cyan.
guidLights.color.magenta	boolean		The light can be magenta.
guidLights.color.white	boolean		The light can be white.

Name	Type	Default	Description
guidLights.direction	object		Indicates which directions are supported by the guidelight.
guidLights.direction.entry	boolean		The light can indicate entry.
guidLights.direction.exit	boolean		The light can indicate exit.
powerSaveControl	boolean		Specifies whether power saving control is available
antiFraudModule	boolean		Specifies whether the anti-fraud module is available
synchronizableCommands	array		list of commands support synchronization.
	object		
type	string		Supplies the type of Dispenser. Following values are possible: "tellerBill": The Dispenser is a Teller Bill Dispenser. "selfServiceBill": The Dispenser is a Self-Service Bill Dispenser. "tellerCoin": The Dispenser is a Teller Coin Dispenser. "selfServiceCoin": The Dispenser is a Self-Service Coin Dispenser.
maxDispenseItems	integer		Supplies the maximum number of items that can be dispensed in a single dispense operation. If no limit applies this value will be zero - in this case, if an attempt is made to dispense more items than the hardware limitations will allow, the Service will implement the dispense as a series of sub-dispense operations (see section Sub-Dispensing Command Flow).

Name	Type	Default	Description
shutter	boolean		Specifies whether or not the commands Dispenser.OpenShutter and Dispenser.CloseShutter are supported.
shutterControl	boolean		If set to TRUE the shutter is controlled implicitly by the Service. If set to FALSE the shutter must be controlled explicitly by the application using the Dispenser.OpenShutter and the Dispenser.CloseShutter commands. This field is always set to TRUE if the device has no shutter. This field applies to all shutters and all output positions.
retractAreas	object		Specifies the area to which items may be retracted. If the device does not have a retract capability all flags will be set to false.
retractAreas.retract	boolean		The items may be retracted to a retract cash unit.
retractAreas.transport	boolean		The items may be retracted to the transport.
retractAreas.stacker	boolean		The items may be retracted to the intermediate stacker.
retractAreas.reject	boolean		The items may be retracted to a reject cash unit.
retractAreas.itemCassette	boolean		The items may be retracted to the item cassettes, i.e. cassettes that can be dispensed from.
retractTransportActions	object		Specifies the actions which may be performed on items which have been retracted to the transport. If the device does not have the capability to retract items to the transport or move items from the transport all flags will be set to false.

Name	Type	Default	Description
retractTransportActions.present	boolean		The items may be presented.
retractTransportActions.retract	boolean		The items may be moved to a retract cash unit.
retractTransportActions.reject	boolean		The items may be moved to a reject bin.
retractTransportActions.itemCassette	boolean		The items may be moved to the item cassettes, i.e. cassettes that can be dispensed from.
retractStackerActions	object		Specifies the actions which may be performed on items which have been retracted to the stacker. If the device does not have the capability to retract items to the stacker or move items from the stacker all flags will be set to false.
retractStackerActions.present	boolean		The items may be presented.
retractStackerActions.retract	boolean		The items may be moved to a retract cash unit.
retractStackerActions.reject	boolean		The items may be moved to a reject bin.
retractStackerActions.itemCassette	boolean		The items may be moved to the item cassettes, i.e. cassettes that can be dispensed from.
safeDoor	boolean		Specifies whether or not the Dispenser.OpenSafeDoor command is supported.
cashBox	boolean		This field is only applicable to Dispenser types "tellerBill" and "tellerCoin". It specifies whether or not tellers have been assigned a cash box.

Name	Type	Default	Description
intermediateStacker	boolean		Specifies whether or not the Dispenser supports stacking items to an intermediate position before the items are moved to the exit position. If this value is TRUE, the field "present" of the Dispenser.Dispense command can be set to FALSE.
itemsTakenSensor	boolean		Specifies whether the Dispenser can detect when items at the exit position are taken by the user. If set to TRUE the Service generates an accompanying Dispenser.ItemsTakenEvent. If set to FALSE this event is not generated. This field applies to all output positions.
positions	object		Specifies the Dispenser output positions which are available.
positions.left	boolean		The Dispenser has a left output position.
positions.right	boolean		The Dispenser has a right output position.
positions.center	boolean		The Dispenser has a center output position.
positions.top	boolean		The Dispenser has a top output position.
positions.bottom	boolean		The Dispenser has a bottom output position.
positions.front	boolean		The Dispenser has a front output position.
positions.rear	boolean		The Dispenser has a rear output position.
moveItems	object		Specifies the Dispenser move item options which are available.

Name	Type	Default	Description
moveItems.fromCashUnit	boolean		The Dispenser can dispense items from the cash units to the intermediate stacker while there are items on the transport.
moveItems.toCashUnit	boolean		The Dispenser can retract items to the cash units while there are items on the intermediate stacker.
moveItems.toTransport	boolean		The Dispenser can retract items to the transport while there are items on the intermediate stacker.
moveItems.toStacker	boolean		The Dispenser can dispense items from the cash units to the intermediate stacker while there are already items on the intermediate stacker that have not been in customer access. Items remaining on the stacker from a previous dispense may first need to be rejected explicitly by the application if they are not to be presented.
exchangeType	object		Specifies the type of cash unit exchange operations supported by the Dispenser.
exchangeType.byHand	boolean		The Dispenser supports manual replenishment either by filling the cash unit by hand or by replacing the cash unit.
exchangeType.toCassettes	boolean		The Dispenser supports moving items from the replenishment cash unit to another cash unit.

Name	Type	Default	Description
prepareDispense	boolean		On some hardware it can take a significant amount of time for the dispenser to get ready to dispense media. On this type of hardware the Dispenser.PrepareDispense command can be used to improve transaction performance. This flag indicates if the hardware requires the application to use the Dispenser.PrepareDispense command to maximize transaction performance. If this flag is TRUE then the Dispenser.PrepareDispense command is supported and can be used to improve transaction performance. If this flag is FALSE then the Dispenser.PrepareDispense command is not supported.
itemInfoTypes	object		Specifies the types of information that can be retrieved through the CashManagement.GetItemInfo command.
itemInfoTypes.serialNumber	boolean		Serial Number of the item.
itemInfoTypes.signature	boolean		Signature of the item.
itemInfoTypes.imageFile	boolean		Image file of the item.
blacklist	boolean		Specifies whether the device has the capability to maintain a blacklist of serial numbers as well as supporting the associated operations. This can either be TRUE if the device has the capability or FALSE if it does not.

Name	Type	Default	Description
classificationList	boolean		Specifies whether the device has the capability to maintain a classification list of serial numbers as well as supporting the associated operations. This can either be TRUE if the device has the capability or FALSE if it does not.

Example Message (generated)

```
{
  "headers": {
    "requestId": "b34800d0-9dd2-4d50-89ea-92d1b13df54b",
    "type": "command",
    "name": "string"
  },
  "payload": {
    "status": "ok",
    "errorDescription": "string",
    "class": "IDC",
    "compound": true,
    "extra": [
      "string"
    ],
    "guidLights": [
      {
        "flashRate": {
          "slow": true,
          "medium": true,
          "quick": true,
          "continuous": true
        },
        "color": {
          "red": true,
          "green": true,
          "yellow": true,
          "blue": true,
          "cyan": true,
          "magenta": true,
          "white": true
        },
        "direction": {
          "entry": true,
          "exit": true
        }
      }
    ],
    "powerSaveControl": true,
    "antiFraudModule": true,
    "synchronizableCommands": [
      "string"
    ],
    "type": "tellerBill",
    "maxDispenseItems": 0,
    "shutter": true,
    "shutterControl": true,
    "retractAreas": {
      "retract": true,
      "transport": true,
      "stacker": true,
      "reject": true,
      "itemCassette": true
    },
    "retractTransportActions": {
      "present": true,
      "retract": true,
      "reject": true,
      "itemCassette": true
    },
    "retractStackerActions": {
      "present": true,
      "retract": true,
      "reject": true,
      "itemCassette": true
    },
    "safeDoor": true,
    "cashBox": true,
    "intermediateStacker": true,
    "itemsTakenSensor": true,
    "positions": {
      "left": true,
      "right": true,
      "center": true,
      "top": true,
      "bottom": true,
      "front": true,
      "rear": true
    },
    "moveItems": {
      "fromCashUnit": true,
      "toCashUnit": true,
      "toTransport": true,
      "toStacker": true
    },
    "exchangeType": {
      "byHand": true,
      "toCassettes": true
    },
    "prepareDispense": true,
    "itemInfoTypes": {
      "serialNumber": true,
      "signature": true,
      "imageFile": true
    },
    "blacklist": true,
    "classificationList": true
  }
}
```

Event Messages

Dispenser.MixTypes

Description

This command is used to obtain a list of supported mix algorithms and available house mix tables.

Command Message

Message Header

Name	Type	Default	Description
requestId (Required)	string		Unique request identifier supplied by the client used to correlate the command with responses, events and completions. For Unsolicited Events the field will be empty.
type (Required)	string		The message type, either command, response, event or completion.
name (Required)	string		The original message name, for example "CardReader.Status"

Message Payload

Name	Type	Default	Description
timeout	integer	0	Timeout in milliseconds for the command to complete. If set to zero, the command will not timeout but can be cancelled.

Example Message (generated)

```
{
  "headers": {
    "requestId": "b34800d0-9dd2-4d50-89ea-92d1b13df54b",
    "type": "command",
    "name": "string"
  },
  "payload": {
    "timeout": "5000"
  }
}
```

Completion Message

Message Header

Name	Type	Default	Description
requestId (Required)	string		Unique request identifier supplied by the client used to correlate the command with responses, events and completions. For Unsolicited Events the field will be empty.
type (Required)	string		The message type, either command, response, event or completion.
name (Required)	string		The original message name, for example "CardReader.Status"

Message Payload

Name	Type	Default	Description
status	string		ok if the commmand was successful otherwise error
errorDescription	string		If error, identified that cause of the error
mixTypes	array		Array of mix type objects.
mixTypes.mixNumber	integer		Number identifying the mix algorithm or the house mix table. This number can be passed to the Dispenser.MixTable, Dispenser.Dispense and Dispenser.Denominate commands.
mixTypes.mixType	string		Specifies whether the mix type is an algorithm or a house mix table. Possible values are "mixAlgorithm" and "mixTable".
mixTypes.subType	integer		Contains a vendor-defined number that identifies the type of algorithm. Individual vendor-defined mix algorithms are defined above hexadecimal 7FFF. Mix algorithms which are provided by the Service are in the range hexadecimal 8000 - 8FFF. Application defined mix algorithms start at hexadecimal 9000. All numbers below 8000 hexadecimal are reserved. If *mixType* is "mixTable", this value will be zero.
mixTypes.name	string		Name of the table/algorithm used.

Example Message (generated)

```
{
  "headers": {
    "requestId": "b34800d0-9dd2-4d50-89ea-92d1b13df54b",
    "type": "command",
    "name": "string"
  },
  "payload": {
    "status": "ok",
    "errorDescription": "string",
    "mixTypes": [
      {
        "mixNumber": 0,
        "mixType": "mixAlgorithm",
        "subType": 0,
        "name": "string"
      }
    ]
  }
}
```

Event Messages

Dispenser.MixTable

Description

This command is used to obtain the house mix table specified by the supplied mix number.

Command Message

Message Header

Name	Type	Default	Description
requestId (Required)	string		Unique request identifier supplied by the client used to correlate the command with responses, events and completions. For Unsolicited Events the field will be empty.
type (Required)	string		The message type, either command, response, event or completion.
name (Required)	string		The original message name, for example "CardReader.Status"

Message Payload

Name	Type	Default	Description
timeout	integer	0	Timeout in milliseconds for the command to complete. If set to zero, the command will not timeout but can be cancelled.
mixNumber	integer		Number of the requested house mix table.

Example Message (generated)

```
{
  "headers": {
    "requestId": "b34800d0-9dd2-4d50-89ea-92d1b13df54b",
    "type": "command",
    "name": "string"
  },
  "payload": {
    "timeout": "5000",
    "mixNumber": 0
  }
}
```

Completion Message

Message Header

Name	Type	Default	Description
requestId (Required)	string		Unique request identifier supplied by the client used to correlate the command with responses, events and completions. For Unsolicited Events the field will be empty.
type (Required)	string		The message type, either command, response, event or completion.
name (Required)	string		The original message name, for example "CardReader.Status"

Message Payload

Name	Type	Default	Description
status	string		ok if the commmand was successful otherwise error
errorDescription	string		If error, identified that cause of the error
	object		
mixNumber	integer		Number identifying the house mix table.
name	string		Name of the house mix table.
mixHeader	array		Array of integers; each element defines the value of the item corresponding to its respective column (see section CashManagement.CurrencyExp).
mixRows	array		Array of rows of the mix table.
mixRows.amount	integer		Amount denominated by this mix row (see section CashManagement.CurrencyExp).
mixRows.mixture	array		A mix row, an array of integers; each element defines the quantity of each item denomination in the mix used in the denomination of *amount*. The value of each array element is defined by the *mixHeader*.

Example Message (generated)

```
{
  "headers": {
    "requestId": "b34800d0-9dd2-4d50-89ea-92d1b13df54b",
    "type": "command",
    "name": "string"
  },
  "payload": {
    "status": "ok",
    "errorDescription": "string",
    "mixNumber": 0,
    "name": "string",
    "mixHeader": [
      0
    ],
    "mixRows": [
      {
        "amount": 0,
        "mixture": [
          0
        ]
      }
    ]
  }
}
```

Event Messages

Dispenser.PresentStatus

Description

This command is used to obtain the status of the most recent attempt to dispense and/or present items to the customer from a specified output position. The items may have been dispensed and/or presented as a result of the Dispenser.Present or Dispenser.Dispense command. This status is not updated as a result of any other command that can dispense/present items.

This value is persistent and is valid until the next time an attempt is made to present or dispense items to the customer.

The denominations reported by this command may not accurately reflect the operation if the cash units have been re-configured (e.g. if the values associated with a cash unit are changed, or new cash units are configured).

Command Message

Message Header

Name	Type	Default	Description
requestId (Required)	string		Unique request identifier supplied by the client used to correlate the command with responses, events and completions. For Unsolicited Events the field will be empty.
type (Required)	string		The message type, either command, response, event or completion.
name (Required)	string		The original message name, for example "CardReader.Status"

Message Payload

Name	Type	Default	Description
timeout	integer	0	Timeout in milliseconds for the command to complete. If set to zero, the command will not timeout but can be cancelled.

Name	Type	Default	Description
position	string		Required output position. Following values are possible: "default": The default configuration. "left": The left output position. "right": The right output position. "center": The center output position. "top": The top output position. "bottom": The bottom output position. "front": The front output position. "rear": The rear output position.

Example Message (generated)

```
{
  "headers": {
    "requestId": "b34800d0-9dd2-4d50-89ea-92d1b13df54b",
    "type": "command",
    "name": "string"
  },
  "payload": {
    "timeout": "5000",
    "position": "default"
  }
}
```

Completion Message

Message Header

Name	Type	Default	Description
requestId (Required)	string		Unique request identifier supplied by the client used to correlate the command with responses, events and completions. For Unsolicited Events the field will be empty.
type (Required)	string		The message type, either command, response, event or completion.
name (Required)	string		The original message name, for example "CardReader.Status"

Message Payload

Name	Type	Default	Description
status	string		ok if the commmand was successful otherwise error
errorDescription	string		If error, identified that cause of the error
	object		
denomination	object		Denomination structure which contains the amount dispensed from the specified output position and the number of items dispensed from each cash unit. Where the capability *moveItems* reports *toStacker* this value is cumulative across a series of Dispenser.Dispense calls that add additional items to the stacker. Where mixed currencies were dispensed the *amount* field in the returned denomination structure will be zero and the *currencyID* field will be omitted.
denomination.currencyID	string		Identification of currency in ISO format [Ref. 2]. Where the denomination contains multiple currencies this field should be omitted.
denomination.amount	integer		The amount to be denominated or dispensed. Where the denomination contains multiple currencies this value is zero.

Name	Type	Default	Description
denomination.values	array		This list specifies the number of items to take from each of the cash units. This list corresponds to the array of cash unit structures returned by the last CashManagement.CashUnitInfo command or set by the last CashManagement.SetCashUnitInfo or CashManagement.EndExchange commands. The first value in the array is related to the cash structure with the index number 1. This array contains a field for each possible cash unit. If a cash unit is not required in the denomination its corresponding field in this array should be set to zero. If the application does not wish to specify a denomination, it should omit the values property.
denomination.cashBox	integer		Only applies to Teller Dispensers. Amount to be paid from the teller's cash box.
presentState	string		Supplies the status of the last dispense or present operation. Following values are possible: "presented": The items were presented. This status is set as soon as the customer has access to the items. "notPresented": The customer has not had access to the items. "unknown": It is not known if the customer had access to the items.
extra	array		Pointer to a list of vendor-specific, or any other extended, information. The information is returned as a series of “key=value” strings so that it is easily extensible by Service Providers.

Example Message (generated)

```
{
  "headers": {
    "requestId": "b34800d0-9dd2-4d50-89ea-92d1b13df54b",
    "type": "command",
    "name": "string"
  },
  "payload": {
    "status": "ok",
    "errorDescription": "string",
    "denomination": {
      "currencyID": "string",
      "amount": 0,
      "values": [
        0
      ],
      "cashBox": 0
    },
    "presentState": "presented",
    "extra": [
      "string"
    ]
  }
}
```

Event Messages

Dispenser.Denominate

Description

This command provides a denomination. A denomination specifies the number of items which are required from each cash unit in order to satisfy a given amount. The denomination depends upon the currency, the mix algorithm and any partial denomination supplied by the application.

This command can also be used to validate that any denomination supplied by the application can be dispensed.

If items of differing currencies are to be included in the same denomination then the currency field must be omitted, the amount must be zero and the mix number must be 0 ("individual"). However, these restrictions do not apply if a single currency is combined with non-currency items, such as coupons.

If the *cashBox* field returned by the Dispenser.Capabilites command is TRUE then, if the entire denomination cannot be satisfied, a partial denomination will be returned with the remaining amount to be supplied from the teller's cash box.

This command can be used in four different ways:

1. In order to check that it is possible to dispense a given denomination. The input parameters to the command are currency and denomination, with a mix number of 0 ("individual") and an amount of zero. If items of differing currencies are to be dispensed then the currency field should be omitted.

2. In order to validate that a given amount matches a given denomination and that it is possible to dispense the denomination. The input parameters to the command should be amount, currency and denomination, with a mix number of 0 ("individual").

3. In order to obtain a denomination of a given amount. The input parameters supplied should be amount, currency and mix number.

4. In order to complete a partial denomination of a given amount. In this case the input parameters to the command should be currency, amount, mix number and either a partially specified denomination or a minimum amount from the cash box. A completed denomination is returned. *cashBox* of the denomination structure may be updated as a result of this command.

Command Message

Message Header

Name	Type	Default	Description
requestId (Required)	string		Unique request identifier supplied by the client used to correlate the command with responses, events and completions. For Unsolicited Events the field will be empty.
type (Required)	string		The message type, either command, response, event or completion.
name (Required)	string		The original message name, for example "CardReader.Status"

Message Payload

Name	Type	Default	Description
timeout	integer	0	Timeout in milliseconds for the command to complete. If set to zero, the command will not timeout but can be cancelled.
	object		
tellerID	integer		Identification of teller. This field is ignored if the device is a Self-Service Dispenser.
mixNumber	integer		Mix algorithm or house mix table to be used.
denomination	object		Denomination object describing the contents of the denomination operation.
denomination.currencyID	string		Identification of currency in ISO format [Ref. 2]. Where the denomination contains multiple currencies this field should be omitted.
denomination.amount	integer		The amount to be denominated or dispensed. Where the denomination contains multiple currencies this value is zero.
denomination.values	array		This list specifies the number of items to take from each of the cash units. This list corresponds to the array of cash unit structures returned by the last CashManagement.CashUnitInfo command or set by the last CashManagement.SetCashUnitInfo or CashManagement.EndExchange commands. The first value in the array is related to the cash structure with the index number 1. This array contains a field for each possible cash unit. If a cash unit is not required in the denomination its corresponding field in this array should be set to zero. If the application does not wish to specify a denomination, it should omit the values property.
denomination.cashBox	integer		Only applies to Teller Dispensers. Amount to be paid from the teller’s cash box.

Example Message (generated)

```
{
  "headers": {
    "requestId": "b34800d0-9dd2-4d50-89ea-92d1b13df54b",
    "type": "command",
    "name": "string"
  },
  "payload": {
    "timeout": "5000",
    "tellerID": 0,
    "mixNumber": 0,
    "denomination": {
      "currencyID": "string",
      "amount": 0,
      "values": [
        0
      ],
      "cashBox": 0
    }
  }
}
```

Completion Message

Message Header

Name	Type	Default	Description
requestId (Required)	string		Unique request identifier supplied by the client used to correlate the command with responses, events and completions. For Unsolicited Events the field will be empty.
type (Required)	string		The message type, either command, response, event or completion.
name (Required)	string		The original message name, for example "CardReader.Status"

Message Payload

Name	Type	Default	Description
status	string		ok if the commmand was successful otherwise error
errorDescription	string		If error, identified that cause of the error
	object		
currencyID	string		Identification of currency in ISO format [Ref. 2]. Where the denomination contains multiple currencies this field should be omitted.
amount	integer		The amount to be denominated or dispensed. Where the denomination contains multiple currencies this value is zero.

Name	Type	Default	Description
values	array		This list specifies the number of items to take from each of the cash units. This list corresponds to the array of cash unit structures returned by the last CashManagement.CashUnitInfo command or set by the last CashManagement.SetCashUnitInfo or CashManagement.EndExchange commands. The first value in the array is related to the cash structure with the index number 1. This array contains a field for each possible cash unit. If a cash unit is not required in the denomination its corresponding field in this array should be set to zero. If the application does not wish to specify a denomination, it should omit the values property.
cashBox	integer		Only applies to Teller Dispensers. Amount to be paid from the teller’s cash box.

Example Message (generated)

```
{
  "headers": {
    "requestId": "b34800d0-9dd2-4d50-89ea-92d1b13df54b",
    "type": "command",
    "name": "string"
  },
  "payload": {
    "status": "ok",
    "errorDescription": "string",
    "currencyID": "string",
    "amount": 0,
    "values": [
      0
    ],
    "cashBox": 0
  }
}
```

Event Messages

- CashManagement.CashUnitErrorEvent

Dispenser.Dispense

Description

This command performs the dispensing of items to the customer. The command provides the same functionality as the Dispenser.Denominate command plus the additional functionality of dispensing the items. If items of differing currencies are to be dispensed then the currency field must be omitted, the amount must be zero and the mix number must be 0 ("individual"). However, these restrictions do not apply if a single currency is dispensed with non-currency items, such as coupons.

The Dispenser.Dispense command can be used in the following ways:

1. The input parameters to the command are amount, currency and denomination. The mix number is 0 ("individual"). In this case, the denomination is checked for validity and, if valid, is dispensed.

2. The input parameters are amount, currency and mix number. In this case the amount is denominated and, if this succeeds, the items are dispensed.

3. If the amount is zero, but the currency and the denomination are supplied with a mix number of 0 ("individual") the denomination is checked for validity and, if valid, is dispensed.

4. The command will calculate a partial denomination of a given amount and dispense the complete denomination. In this case the input parameters to the command should be currency, amount, mix number and either a partially specified denomination or a minimum amount from the cash box. The cash box amount may be updated as a result of this command.

When more than one physical cash unit exists for a logical cash unit number, the device selects the actual physical cash unit to use in the dispense operation.

If the *cashBox* field returned by the Dispenser.Capabilities command is TRUE then, if the entire denomination cannot be satisfied, a partial denomination will be returned with the remaining amount to be supplied from the teller's cash box.

If the device is a Teller Dispenser, the input field *position* can be set to "default". If this is the case the *tellerID* is used to perform the dispense operation to the assigned teller position.

The field *present* of the Dispenser.Dispense command message determines whether items are actually presented to the user as part of the dispense operation. If this field is set to TRUE then the items will be moved to the exit slot, if it is FALSE the items will be moved to an intermediate stacker. In the second case it will be necessary to use the Dispenser.Present command to present the items to the user. If *present* is set to FALSE then the *position* field is ignored. If the Dispenser does not have an intermediate stacker then *present* is ignored.

If *present* is set to TRUE and a shutter exists, then it will be implicitly controlled during the present operation, even if the *shutterControl* capability is set to FALSE. The shutter will be closed when the user removes the items or the items are retracted.

Note that a level 4 note can be dispensed, but is not necessarily presented to the customer. e.g. a note can be skewed, or can be unfit for dispensing.

The values in the completion messsage report the amount dispensed and the number of items dispensed from each cash unit.

Command Message

Message Header

Name	Type	Default	Description
requestId (Required)	string		Unique request identifier supplied by the client used to correlate the command with responses, events and completions. For Unsolicited Events the field will be empty.
type (Required)	string		The message type, either command, response, event or completion.
name (Required)	string		The original message name, for example "CardReader.Status"

Message Payload

Name	Type	Default	Description
timeout	integer	0	Timeout in milliseconds for the command to complete. If set to zero, the command will not timeout but can be cancelled.
	object		

Name	Type	Default	Description
tellerID	integer		Identifies the teller. This field is ignored if the device is a Self-Service Dispenser.
mixNumber	integer		Mix algorithm or house mix table to be used to create a denomination of the supplied amount. If the value is 0 ("individual"), the denomination supplied in the *denomination* field is validated prior to the dispense operation. If it is found to be invalid no alternative denomination will be calculated.
position	string		Required output position. Following values are possible: "default": The default configuration information is used. This can be either position dependent or teller dependent. "left": Present items to left side of device. "right": Present items to right side of device. "center": Present items to center output position. "top": Present items to the top output position. "bottom": Present items to the bottom output position. "front": Present items to the front output position. "rear": Present items to the rear output position.
present	boolean		If this field is set to TRUE then the items will be moved to the exit slot, if it is FALSE the items will be moved to an intermediate stacker.
denomination	object		Denomination object describing the denominations used for the dispense operation.

Name	Type	Default	Description
denomination.currencyID	string		Identification of currency in ISO format [Ref. 2]. Where the denomination contains multiple currencies this field should be omitted.
denomination.amount	integer		The amount to be denominated or dispensed. Where the denomination contains multiple currencies this value is zero.
denomination.values	array		This list specifies the number of items to take from each of the cash units. This list corresponds to the array of cash unit structures returned by the last CashManagement.CashUnitInfo command or set by the last CashManagement.SetCashUnitInfo or CashManagement.EndExchange commands. The first value in the array is related to the cash structure with the index number 1. This array contains a field for each possible cash unit. If a cash unit is not required in the denomination its corresponding field in this array should be set to zero. If the application does not wish to specify a denomination, it should omit the values property.
denomination.cashBox	integer		Only applies to Teller Dispensers. Amount to be paid from the teller's cash box.

Example Message (generated)

```
{
  "headers": {
    "requestId": "b34800d0-9dd2-4d50-89ea-92d1b13df54b",
    "type": "command",
    "name": "string"
  },
  "payload": {
    "timeout": "5000",
    "tellerID": 0,
    "mixNumber": 0,
    "position": "default",
    "present": true,
    "denomination": {
      "currencyID": "string",
      "amount": 0,
      "values": [
        0
      ],
      "cashBox": 0
    }
  }
}
```

Completion Message

Message Header

Name	Type	Default	Description
requestId (Required)	string		Unique request identifier supplied by the client used to correlate the command with responses, events and completions. For Unsolicited Events the field will be empty.
type (Required)	string		The message type, either command, response, event or completion.
name (Required)	string		The original message name, for example "CardReader.Status"

Message Payload

Name	Type	Default	Description
status	string		ok if the commmand was successful otherwise error
errorDescription	string		If error, identified that cause of the error
	object		
currencyID	string		Identification of currency in ISO format [Ref. 2]. Where the denomination contains multiple currencies this field should be omitted.
amount	integer		The amount to be denominated or dispensed. Where the denomination contains multiple currencies this value is zero.

Name	Type	Default	Description
values	array		This list specifies the number of items to take from each of the cash units. This list corresponds to the array of cash unit structures returned by the last CashManagement.CashUnitInfo command or set by the last CashManagement.SetCashUnitInfo or CashManagement.EndExchange commands. The first value in the array is related to the cash structure with the index number 1. This array contains a field for each possible cash unit. If a cash unit is not required in the denomination its corresponding field in this array should be set to zero. If the application does not wish to specify a denomination, it should omit the values property.
cashBox	integer		Only applies to Teller Dispensers. Amount to be paid from the teller's cash box.

Example Message (generated)

```
{
  "headers": {
    "requestId": "b34800d0-9dd2-4d50-89ea-92d1b13df54b",
    "type": "command",
    "name": "string"
  },
  "payload": {
    "status": "ok",
    "errorDescription": "string",
    "currencyID": "string",
    "amount": 0,
    "values": [
      0
    ],
    "cashBox": 0
  }
}
```

Event Messages

- CashManagement.CashUnitThresholdEvent
- Dispenser.DelayedDispenseEvent
- Dispenser.StartDispenseEvent
- CashManagement.CashUnitErrorEvent
- Dispenser.ItemsTakenEvent
- Dispenser.PartialDispenseEvent
- Dispenser.SubDispenseOkEvent
- Dispenser.IncompleteDispenseEvent
- CashManagement.NoteErrorEvent
- CashManagement.InputP6Event
- CashManagement.InfoAvailableEvent
- Dispenser.ShutterStatusChangedEvent

Dispenser.Present

Description

This command will move items to the exit position for removal by the user. If a shutter exists, then it will be implicitly controlled during the present operation, even if the *shutterControl* capability is set to FALSE. The shutter will be closed when the user removes the items or the items are retracted. If *position* is "default" the position set in the Dispenser.Dispense command which caused these items to be dispensed will be used.

When this command successfully completes the items are in customer access.

Command Message

Message Header

Name	Type	Default	Description
requestId (Required)	string		Unique request identifier supplied by the client used to correlate the command with responses, events and completions. For Unsolicited Events the field will be empty.
type (Required)	string		The message type, either command, response, event or completion.
name (Required)	string		The original message name, for example "CardReader.Status"

Message Payload

Name	Type	Default	Description
timeout	integer	0	Timeout in milliseconds for the command to complete. If set to zero, the command will not timeout but can be cancelled.
position	string		Output position where the amount is to be presented. Following values are possible: "default": The default configuration. "left": The left output position. "right": The right output position. "center": The center output position. "top": The top output position. "bottom": The bottom output position. "front": The front output position. "rear": The rear output position.

Example Message (generated)

```
{
  "headers": {
    "requestId": "b34800d0-9dd2-4d50-89ea-92d1b13df54b",
    "type": "command",
    "name": "string"
  },
  "payload": {
    "timeout": "5000",
    "position": "default"
  }
}
```

Completion Message

Message Header

Name	Type	Default	Description
requestId (Required)	string		Unique request identifier supplied by the client used to correlate the command with responses, events and completions. For Unsolicited Events the field will be empty.
type (Required)	string		The message type, either command, response, event or completion.
name (Required)	string		The original message name, for example "CardReader.Status"

Message Payload

Name	Type	Default	Description
status	string		ok if the commmand was successful otherwise error
errorDescription	string		If error, identified that cause of the error

Example Message (generated)

```
{
  "headers": {
    "requestId": "b34800d0-9dd2-4d50-89ea-92d1b13df54b",
    "type": "command",
    "name": "string"
  },
  "payload": {
    "status": "ok",
    "errorDescription": "string"
  }
}
```

Event Messages

- CashManagement.CashUnitThresholdEvent
- Dispenser.ItemsTakenEvent
- CashManagement.InputP6Event
- CashManagement.InfoAvailableEvent

- Dispenser.ShutterStatusChangedEvent

Dispenser.Reject

Description

This command will move items from the intermediate stacker and transport them to a reject cash unit (i.e. a cash unit with *type* "rejectCassette"). The *count* field of the reject cash unit is incremented by the number of items that were thought to be present at the time of the reject or the number counted by the device during the reject. Note that the reject bin count is unreliable.

Command Message

Message Header

Name	Type	Default	Description
requestId (Required)	string		Unique request identifier supplied by the client used to correlate the command with responses, events and completions. For Unsolicited Events the field will be empty.
type (Required)	string		The message type, either command, response, event or completion.
name (Required)	string		The original message name, for example "CardReader.Status"

Message Payload

Name	Type	Default	Description
timeout	integer	0	Timeout in milliseconds for the command to complete. If set to zero, the command will not timeout but can be cancelled.

Example Message (generated)

```
{
  "headers": {
    "requestId": "b34800d0-9dd2-4d50-89ea-92d1b13df54b",
    "type": "command",
    "name": "string"
  },
  "payload": {
    "timeout": "5000"
  }
}
```

Completion Message

Message Header

Name	Type	Default	Description
requestId (Required)	string		Unique request identifier supplied by the client used to correlate the command with responses, events and completions. For Unsolicited Events the field will be empty.

Name	Type	Default	Description
type (Required)	string		The message type, either command, response, event or completion.
name (Required)	string		The original message name, for example "CardReader.Status"

Message Payload

Name	Type	Default	Description
status	string		ok if the commmand was successful otherwise error
errorDescription	string		If error, identified that cause of the error

Example Message (generated)

```
{
  "headers": {
    "requestId": "b34800d0-9dd2-4d50-89ea-92d1b13df54b",
    "type": "command",
    "name": "string"
  },
  "payload": {
    "status": "ok",
    "errorDescription": "string"
  }
}
```

Event Messages

- CashManagement.CashUnitThresholdEvent
- CashManagement.CashUnitErrorEvent
- CashManagement.InputP6Event
- CashManagement.InfoAvailableEvent

Dispenser.Retract

Description

This command will retract items which may have been in customer access from an output position or from internal areas within the Dispenser. Retracted items will be moved to either a retract cash unit, a reject cash unit, item cash units, the transport or the intermediate stacker. After the items are retracted the shutter is closed automatically, even if the *shutterControl* capability is set to FALSE.

If items are moved to a retract cash unit (i.e. a cash unit with *type* "retractCassette"), then the *count* field of the retract cash unit must be incremented by 1 to specify the number of retracts. If items are moved to any other cash unit (e.g. a cash unit with *type* "rejectCassette") then the *count* field of the cash unit must be incremented by the number of items that were thought to be present at the time the Dispenser.Retract command was issued or the number counted by the device during the retract. Note that reject bin counts are unreliable.

Command Message

Message Header

Name	Type	Default	Description
requestId (Required)	string		Unique request identifier supplied by the client used to correlate the command with responses, events and completions. For Unsolicited Events the field will be empty.
type (Required)	string		The message type, either command, response, event or completion.
name (Required)	string		The original message name, for example "CardReader.Status"

Message Payload

Name	Type	Default	Description
timeout	integer	0	Timeout in milliseconds for the command to complete. If set to zero, the command will not timeout but can be cancelled.
	object		
outputPosition	string		Output position from which to retract the items. Following values are possible: "default": The default configuration information should be used. "left": Retract items from the left output position. "right": Retract items from the right output position. "center": Retract items from the center output position. "top": Retract items from the top output position. "bottom": Retract items from the bottom output position. "front": Retract items from the front output position. "rear": Retract items from the rear output position.

Name	Type	Default	Description
retractArea	string		This value specifies the area to which the items are to be retracted. Following values are possible: "retract": Retract the items to a retract cash unit. "transport": Retract the items to the transport. "stacker": Retract the items to the intermediate stacker area. "reject": Retract the items to a reject cash unit. "itemCassette": Retract the items to the item cassettes, i.e. cassettes that can be dispensed from.
index	integer		If *retractArea* is set to "retract" this field defines the position inside the retract cash units into which the cash is to be retracted. *index* starts with a value of one (1) for the first retract position and increments by one for each subsequent position. If there are several logical retract cash units (of type "retractCassette" in command CashManagement.CashUnitInfo), *index* would be incremented from the first position of the first retract cash unit to the last position of the last retract cash unit. The maximum value of *index* is the sum of *maximum* of each retract cash unit. If *retractArea* is not set to "retract" the value of this field is ignored.

Example Message (generated)

```
{
  "headers": {
    "requestId": "b34800d0-9dd2-4d50-89ea-92d1b13df54b",
    "type": "command",
    "name": "string"
  },
  "payload": {
    "timeout": "5000",
    "outputPosition": "default",
    "retractArea": "retract",
    "index": 0
  }
}
```

Completion Message

Message Header

Name	Type	Default	Description
requestId (Required)	string		Unique request identifier supplied by the client used to correlate the command with responses, events and completions. For Unsolicited Events the field will be empty.
type (Required)	string		The message type, either command, response, event or completion.
name (Required)	string		The original message name, for example "CardReader.Status"

Message Payload

Name	Type	Default	Description
status	string		ok if the commmand was successful otherwise error
errorDescription	string		If error, identified that cause of the error
	object		
itemNumber	array		Array of item number objects.
itemNumber.currencyID	string		A three character array storing the ISO format [Ref. 2] Currency ID; if the currency of the item is not known this is omitted.
itemNumber.values	integer		The value of a single item expressed in minimum dispense units; or a zero value if the value of the item is not known.
itemNumber.release	integer		The release of the item. The higher this number is, the newer the release. Zero means that there is only one release or the release is not known. This value has not been standardized and therefore a release number of the same item will not necessarily have the same value in different systems.
itemNumber.count	integer		The count of items of the same type moved to the same destination during the execution of this command.

Name	Type	Default	Description
itemNumber.number	integer		The logical number of the cash unit which received items during the execution of this command. This value will be zero if items were moved to the *retractArea* "transport" or "stacker".

Example Message (generated)

```
{
  "headers": {
    "requestId": "b34800d0-9dd2-4d50-89ea-92d1b13df54b",
    "type": "command",
    "name": "string"
  },
  "payload": {
    "status": "ok",
    "errorDescription": "string",
    "itemNumber": [
      {
        "currencyID": "string",
        "values": 0,
        "release": 0,
        "count": 0,
        "number": 0
      }
    ]
  }
}
```

Event Messages

- CashManagement.CashUnitThresholdEvent
- CashManagement.CashUnitErrorEvent
- Dispenser.ItemsTakenEvent
- CashManagement.InputP6Event
- CashManagement.InfoAvailableEvent
- Dispenser.IncompleteRetractEvent
- Dispenser.ShutterStatusChangedEvent

Dispenser.OpenShutter

Description

This command opens the shutter.

Command Message

Message Header

Name	Type	Default	Description
requestId (Required)	string		Unique request identifier supplied by the client used to correlate the command with responses, events and completions. For Unsolicited Events the field will be empty.

Name	Type	Default	Description
type (Required)	string		The message type, either command, response, event or completion.
name (Required)	string		The original message name, for example "CardReader.Status"

Message Payload

Name	Type	Default	Description
timeout	integer	0	Timeout in milliseconds for the command to complete. If set to zero, the command will not timeout but can be cancelled.
position	string		The output position where the shutter is to be opened. If the application does not need to specify a shutter, this field can be omitted or its contents set to "default". Following values are possible: "default": The default configuration information should be used. "left": Open the shutter at the left output position. "right": Open the shutter at the right output position. "center": Open the shutter at the center output position. "top": Open the shutter at the top output position. "bottom": Open the shutter at the bottom output position. "front": Open the shutter at the front output position. "rear": Open the shutter at the rear output position.

Example Message (generated)

```
{
  "headers": {
    "requestId": "b34800d0-9dd2-4d50-89ea-92d1b13df54b",
    "type": "command",
    "name": "string"
  },
  "payload": {
    "timeout": "5000",
    "position": "default"
  }
}
```

Completion Message

Message Header

Name	Type	Default	Description
requestId (Required)	string		Unique request identifier supplied by the client used to correlate the command with responses, events and completions. For Unsolicited Events the field will be empty.
type (Required)	string		The message type, either command, response, event or completion.
name (Required)	string		The original message name, for example "CardReader.Status"

Message Payload

Name	Type	Default	Description
status	string		ok if the commmand was successful otherwise error
errorDescription	string		If error, identified that cause of the error

Example Message (generated)

```
{
  "headers": {
    "requestId": "b34800d0-9dd2-4d50-89ea-92d1b13df54b",
    "type": "command",
    "name": "string"
  },
  "payload": {
    "status": "ok",
    "errorDescription": "string"
  }
}
```

Event Messages

- Dispenser.ShutterStatusChangedEvent

Dispenser.CloseShutter

Description

This command closes the shutter.

Command Message

Message Header

Name	Type	Default	Description
requestId (Required)	string		Unique request identifier supplied by the client used to correlate the command with responses, events and completions. For Unsolicited Events the field will be empty.
type (Required)	string		The message type, either command, response, event or completion.
name (Required)	string		The original message name, for example "CardReader.Status"

Message Payload

Name	Type	Default	Description
timeout	integer	0	Timeout in milliseconds for the command to complete. If set to zero, the command will not timeout but can be cancelled.

Name	Type	Default	Description
position	string		The output position where the shutter is to be closed. If the application does not need to specify a shutter, this field can be omitted or its contents set to "default". Following values are possible: "default": The default configuration information should be used. "left": Close the shutter at the left output position. "right": Close the shutter at the right output position. "center": Close the shutter at the center output position. "top": Close the shutter at the top output position. "bottom": Close the shutter at the bottom output position. "front": Close the shutter at the front output position. "rear": Close the shutter at the rear output position.

Example Message (generated)

```
{
  "headers": {
    "requestId": "b34800d0-9dd2-4d50-89ea-92d1b13df54b",
    "type": "command",
    "name": "string"
  },
  "payload": {
    "timeout": "5000",
    "position": "default"
  }
}
```

Completion Message

Message Header

Name	Type	Default	Description

Name	Type	Default	Description
requestId (Required)	string		Unique request identifier supplied by the client used to correlate the command with responses, events and completions. For Unsolicited Events the field will be empty.
type (Required)	string		The message type, either command, response, event or completion.
name (Required)	string		The original message name, for example "CardReader.Status"

Message Payload

Name	Type	Default	Description
status	string		ok if the commmand was successful otherwise error
errorDescription	string		If error, identified that cause of the error

Example Message (generated)

```
{
  "headers": {
    "requestId": "b34800d0-9dd2-4d50-89ea-92d1b13df54b",
    "type": "command",
    "name": "string"
  },
  "payload": {
    "status": "ok",
    "errorDescription": "string"
  }
}
```

Event Messages

- Dispenser.ShutterStatusChangedEvent

Dispenser.OpenSafeDoor

Description

This command unlocks the safe door or starts the time delay count down prior to unlocking the safe door, if the device supports it. The command completes when the door is unlocked or the timer has started.

Command Message

Message Header

Name	Type	Default	Description
requestId (Required)	string		Unique request identifier supplied by the client used to correlate the command with responses, events and completions. For Unsolicited Events the field will be empty.

Name	Type	Default	Description
type (Required)	string		The message type, either command, response, event or completion.
name (Required)	string		The original message name, for example "CardReader.Status"

Message Payload

Name	Type	Default	Description
timeout	integer	0	Timeout in milliseconds for the command to complete. If set to zero, the command will not timeout but can be cancelled.

Example Message (generated)

```
{
  "headers": {
    "requestId": "b34800d0-9dd2-4d50-89ea-92d1b13df54b",
    "type": "command",
    "name": "string"
  },
  "payload": {
    "timeout": "5000"
  }
}
```

Completion Message

Message Header

Name	Type	Default	Description
requestId (Required)	string		Unique request identifier supplied by the client used to correlate the command with responses, events and completions. For Unsolicited Events the field will be empty.
type (Required)	string		The message type, either command, response, event or completion.
name (Required)	string		The original message name, for example "CardReader.Status"

Message Payload

Name	Type	Default	Description
status	string		ok if the commmand was successful otherwise error
errorDescription	string		If error, identified that cause of the error

Example Message (generated)

```
{
  "headers": {
    "requestId": "b34800d0-9dd2-4d50-89ea-92d1b13df54b",
    "type": "command",
    "name": "string"
  },
  "payload": {
    "status": "ok",
    "errorDescription": "string"
  }
}
```

Event Messages

Dispenser.SetMixTable

Description

This command is used to set up the mix table specified by the mix number. Mix tables are persistent and are available to all applications in the system. An amount can be specified as different denominations within the mix table. If the amount is specified more than once the Service Provider will attempt to denominate or dispense the first amount in the table. If this does not succeed (e.g. because of a cash unit failure) the Service Provider will attempt to denominate or dispense the next amount in the table. The Service Provider can only dispense amounts which are explicitly mentioned in the mix table.

If a mix number passed in already exists then the information is overwritten with the new information.

Command Message

Message Header

Name	Type	Default	Description
requestId (Required)	string		Unique request identifier supplied by the client used to correlate the command with responses, events and completions. For Unsolicited Events the field will be empty.
type (Required)	string		The message type, either command, response, event or completion.
name (Required)	string		The original message name, for example "CardReader.Status"

Message Payload

Name	Type	Default	Description
timeout	integer	0	Timeout in milliseconds for the command to complete. If set to zero, the command will not timeout but can be cancelled.
	object		
mixNumber	integer		Number identifying the house mix table.
name	string		Name of the house mix table.

Name	Type	Default	Description
mixHeader	array		Array of integers; each element defines the value of the item corresponding to its respective column (see section CashManagement.CurrencyExp).
mixRows	array		Array of rows of the mix table.
mixRows.amount	integer		Amount denominated by this mix row (see section CashManagement.CurrencyExp).
mixRows.mixture	array		A mix row, an array of integers; each element defines the quantity of each item denomination in the mix used in the denomination of *amount*. The value of each array element is defined by the *mixHeader*.

Example Message (generated)

```
{
  "headers": {
    "requestId": "b34800d0-9dd2-4d50-89ea-92d1b13df54b",
    "type": "command",
    "name": "string"
  },
  "payload": {
    "timeout": "5000",
    "mixNumber": 0,
    "name": "string",
    "mixHeader": [
      0
    ],
    "mixRows": [
      {
        "amount": 0,
        "mixture": [
          0
        ]
      }
    ]
  }
}
```

Completion Message

Message Header

Name	Type	Default	Description
requestId (Required)	string		Unique request identifier supplied by the client used to correlate the command with responses, events and completions. For Unsolicited Events the field will be empty.
type (Required)	string		The message type, either command, response, event or completion.
name (Required)	string		The original message name, for example "CardReader.Status"

Message Payload

Name	Type	Default	Description
status	string		ok if the commmand was successful otherwise error
errorDescription	string		If error, identified that cause of the error

Example Message (generated)

```
{
  "headers": {
    "requestId": "b34800d0-9dd2-4d50-89ea-92d1b13df54b",
    "type": "command",
    "name": "string"
  },
  "payload": {
    "status": "ok",
    "errorDescription": "string"
  }
}
```

Event Messages

Dispenser.Reset

Description

This command is used by the application to perform a hardware reset which will attempt to return the Dispenser device to a known good state. This command does not over-ride a lock obtained through Common.Lock (TODO) on another application or service handle.

The device will attempt to move any items found anywhere within the device to the position specified within the command payload. This may not always be possible because of hardware problems.

If items are found inside the device the Dispenser.MediaDetectedEvent will be generated and will inform the application where the items were actually moved to.

If an exchange state is active then this command will end the exchange state (even if this command does not complete successfully).

On a recycling device this command is not accepted if a cash-in transaction is active and will return a "deviceNotReady" error.

If items are moved to a retract cash unit (i.e. a cash unit with *type* "retractCassette"), then the *count* field of the retract cash unit must be incremented by 1 to specify the number of operations that changed the count. If items are moved to any other cash unit (e.g. a cash unit with *type* "rejectCassette), then the *count* field of the cash unit must be incremented either by the number of items that were present at the time the Dispenser.Reset command was issued or the number counted by the device during the Dispenser.Reset command. Note that reject bin counts are unreliable.

Command Message

Message Header

Name	Type	Default	Description

Name	Type	Default	Description
requestId (Required)	string		Unique request identifier supplied by the client used to correlate the command with responses, events and completions. For Unsolicited Events the field will be empty.
type (Required)	string		The message type, either command, response, event or completion.
name (Required)	string		The original message name, for example "CardReader.Status"

Message Payload

Name	Type	Default	Description
timeout	integer	0	Timeout in milliseconds for the command to complete. If set to zero, the command will not timeout but can be cancelled.
	object		
number	integer		If non-zero, this value specifies the *number* (as specified by CashManagement.CashUnitInfo) of the single cash unit to be used for the storage of any items found.
retractArea	object		This field is used if items are to be moved to internal areas of the device, including cash units, the intermediate stacker, or the transport.

Name	Type	Default	Description
retractArea.outputPosition	string		Output position from which to retract the items. Following values are possible: "default": The default configuration information should be used. "left": Retract items from the left output position. "right": Retract items from the right output position. "center": Retract items from the center output position. "top": Retract items from the top output position. "bottom": Retract items from the bottom output position. "front": Retract items from the front output position. "rear": Retract items from the rear output position.
retractArea.retractArea	string		This value specifies the area to which the items are to be retracted. Following values are possible: "retract": Retract the items to a retract cash unit. "transport": Retract the items to the transport. "stacker": Retract the items to the intermediate stacker area. "reject": Retract the items to a reject cash unit. "itemCassette": Retract the items to the item cassettes, i.e. cassettes that can be dispensed from.

Name	Type	Default	Description
retractArea.index	integer		If *retractArea* is set to "retract" this field defines the position inside the retract cash units into which the cash is to be retracted. *index* starts with a value of one (1) for the first retract position and increments by one for each subsequent position. If there are several logical retract cash units (of type "retractCassette" in command CashManagement.CashUnitInfo), *index* would be incremented from the first position of the first retract cash unit to the last position of the last retract cash unit. The maximum value of *index* is the sum of *maximum* of each retract cash unit. If *retractArea* is not set to "retract" the value of this field is ignored.
outputPosition	string		The output position to which items are to be moved. This field is only used if *number* is zero and retractArea is omitted. Following values are possible: "default": The default configuration. "left": The left output position. "right": The right output position. "center": The center output position. "top": The top output position. "bottom": The bottom output position. "front": The front output position. "rear": The rear output position.

Example Message (generated)

```
{
  "headers": {
    "requestId": "b34800d0-9dd2-4d50-89ea-92d1b13df54b",
    "type": "command",
    "name": "string"
  },
  "payload": {
    "timeout": "5000",
    "number": 0,
    "retractArea": {
      "outputPosition": "default",
      "retractArea": "retract",
      "index": 0
    },
    "outputPosition": "default"
  }
}
```

Completion Message

Message Header

Name	Type	Default	Description
requestId (Required)	string		Unique request identifier supplied by the client used to correlate the command with responses, events and completions. For Unsolicited Events the field will be empty.
type (Required)	string		The message type, either command, response, event or completion.
name (Required)	string		The original message name, for example "CardReader.Status"

Message Payload

Name	Type	Default	Description
status	string		ok if the commmand was successful otherwise error
errorDescription	string		If error, identified that cause of the error

Example Message (generated)

```
{
  "headers": {
    "requestId": "b34800d0-9dd2-4d50-89ea-92d1b13df54b",
    "type": "command",
    "name": "string"
  },
  "payload": {
    "status": "ok",
    "errorDescription": "string"
  }
}
```

Event Messages

- CashManagement.CashUnitThresholdEvent
- CashManagement.CashUnitErrorEvent
- Dispenser.MediaDetectedEvent
- Dispenser.ItemsTakenEvent
- CashManagement.InputP6Event
- CashManagement.InfoAvailableEvent
- Dispenser.IncompleteRetractEvent
- Dispenser.ShutterStatusChangedEvent

Dispenser.TestCashUnits

Description

This command is used to test cash units following replenishment. The command payload specifies where items dispensed as a result of this command should be moved to. All physical cash units which are testable (i.e. that have a *status* of "ok" or "low" and no application lock in the logical cash unit associated with the physical cash unit) are tested. If the hardware is able to do so tests are continued even if an error occurs while testing one of the cash units. The command completes with success completion message if the Service successfully manages to test all of the testable cash units regardless of the outcome of the test. This is the case if all testable cash units could be tested and a dispense was possible from at least one of the cash units.

A CashManagement.CashUnitErrorEvent will be sent for any logical cash unit which has one or more physical cash units which cannot be tested or which fail the test, even if the logical cash unit has other physical cash units which are successfully tested. **If all the cash units could not be tested or no cash units are testable then a** "cashUnitError" code will be returned and CashManagement.CashUnitErrorEvents generated for every logical cash unit that encountered a problem. The operation performed to test the cash units is vendor dependent. Items may be dispensed or transported into a reject bin as a result of this command.

If no cash units are testable then a "cashUnitError" code will be returned and CashManagement.CashUnitErrorEvents will be generated for every cash unit.

Command Message

Message Header

Name	Type	Default	Description
requestId (Required)	string		Unique request identifier supplied by the client used to correlate the command with responses, events and completions. For Unsolicited Events the field will be empty.
type (Required)	string		The message type, either command, response, event or completion.
name (Required)	string		The original message name, for example "CardReader.Status"

Message Payload

Name	Type	Default	Description
timeout	integer	0	Timeout in milliseconds for the command to complete. If set to zero, the command will not timeout but can be cancelled.
	object		

Name	Type	Default	Description
number	integer		If non-zero, this value specifies the *number* (as specified by CashManagement.CashUnitInfo) of the single cash unit to be used for the storage of any items found.
retractArea	object		This field is used if items are to be moved to internal areas of the device, including cash units, the intermediate stacker, or the transport.
retractArea.outputPosition	string		Output position from which to retract the items. Following values are possible: "default": The default configuration information should be used. "left": Retract items from the left output position. "right": Retract items from the right output position. "center": Retract items from the center output position. "top": Retract items from the top output position. "bottom": Retract items from the bottom output position. "front": Retract items from the front output position. "rear": Retract items from the rear output position.

Name	Type	Default	Description
retractArea.retractArea	string		This value specifies the area to which the items are to be retracted. Following values are possible: "retract": Retract the items to a retract cash unit. "transport": Retract the items to the transport. "stacker": Retract the items to the intermediate stacker area. "reject": Retract the items to a reject cash unit. "itemCassette": Retract the items to the item cassettes, i.e. cassettes that can be dispensed from.
retractArea.index	integer		If *retractArea* is set to "retract" this field defines the position inside the retract cash units into which the cash is to be retracted. *index* starts with a value of one (1) for the first retract position and increments by one for each subsequent position. If there are several logical retract cash units (of type "retractCassette" in command CashManagement.CashUnitInfo), *index* would be incremented from the first position of the first retract cash unit to the last position of the last retract cash unit. The maximum value of *index* is the sum of *maximum* of each retract cash unit. If *retractArea* is not set to "retract" the value of this field is ignored.

Name	Type	Default	Description
outputPosition	string		The output position to which items are to be moved. This field is only used if *number* is zero and retractArea is omitted. Following values are possible: "default": The default configuration. "left": The left output position. "right": The right output position. "center": The center output position. "top": The top output position. "bottom": The bottom output position. "front": The front output position. "rear": The rear output position.

Example Message (generated)

```
{
  "headers": {
    "requestId": "b34800d0-9dd2-4d50-89ea-92d1b13df54b",
    "type": "command",
    "name": "string"
  },
  "payload": {
    "timeout": "5000",
    "number": 0,
    "retractArea": {
      "outputPosition": "default",
      "retractArea": "retract",
      "index": 0
    },
    "outputPosition": "default"
  }
}
```

Completion Message

Message Header

Name	Type	Default	Description
requestId (Required)	string		Unique request identifier supplied by the client used to correlate the command with responses, events and completions. For Unsolicited Events the field will be empty.
type (Required)	string		The message type, either command, response, event or completion.

Name	Type	Default	Description
name (Required)	string		The original message name, for example "CardReader.Status"

Message Payload

Name	Type	Default	Description
status	string		ok if the commmand was successful otherwise error
errorDescription	string		If error, identified that cause of the error
	object		
list	array		Array of cash unit objects.

Example Message (generated)

```
{
  "headers": {
    "requestId": "b34800d0-9dd2-4d50-89ea-92d1b13df54b",
    "type": "command",
    "name": "string"
  },
  "payload": {
    "status": "ok",
    "errorDescription": "string",
    "list": [
      {
        "number": 0,
        "type": "notApplicable",
        "unitID": "string",
        "currencyID": "string",
        "values": 0,
        "count": 0,
        "maximum": 0,
        "status": "ok",
        "appLock": true,
        "physical": [
          {
            "physicalPositionName": "string",
            "unitID": "string",
            "count": 0,
            "maximum": 0,
            "pStatus": "ok",
            "hardwareSensor": true,
            "initialCount": 0,
            "dispensedCount": 0,
            "presentedCount": 0,
            "retractedCount": 0,
            "rejectCount": 0,
            "cashInCount": 0,
            "extra": [
              "string"
            ]
          }
        ],
        "cashUnitName": "string",
        "cashUnitName": "string",
        "initialCount": 0,
        "dispensedCount": 0,
        "presentedCount": 0,
        "retractedCount": 0,
        "rejectCount": 0,
        "minimum": 0,
        "itemType": {
          "all": true,
          "unfit": true,
          "individual": true,
          "level1": true,
          "level2": true,
          "level3": true,
          "itemProcessor": true,
          "unfitIndividual": true
        },
        "cashInCount": 0,
        "noteNumberList": {
          "noteNumber": [
            {
              "noteID": 0,
              "count": 0
            }
          ]
        },
        "noteIDs": [
          0
        ],
        "extra": [
          "string"
        ]
      }
    ]
  }
}
```

Event Messages

- CashManagement.CashUnitThresholdEvent
- CashManagement.CashUnitErrorEvent
- Dispenser.ItemsTakenEvent
- CashManagement.CashUnitInfoChangedEvent
- CashManagement.NoteErrorEvent
- CashManagement.InputP6Event
- Dispenser.ShutterStatusChangedEvent
- CashManagement.InfoAvailableEvent

Dispenser.Count

Description

This command empties the specified physical cash unit(s). All items dispensed from the cash unit are counted and moved to the specified output location.

The number of items counted can be different from the number of items dispensed in cases where the Dispenser has the ability to detect this information. If the Dispenser cannot differentiate between what is dispensed and what is counted then *dispensed* will be the same as *counted*.

Upon successful Dispenser.Count command execution the physical cash unit(s) *count* field is reset.

Command Message

Message Header

Name	Type	Default	Description
requestId (Required)	string		Unique request identifier supplied by the client used to correlate the command with responses, events and completions. For Unsolicited Events the field will be empty.
type (Required)	string		The message type, either command, response, event or completion.
name (Required)	string		The original message name, for example "CardReader.Status"

Message Payload

Name	Type	Default	Description
timeout	integer	0	Timeout in milliseconds for the command to complete. If set to zero, the command will not timeout but can be cancelled.
	object		
emptyAll	boolean		Specifies whether all physical cash units are to be emptied. If this value is TRUE then physicalPositionName is ignored.

Name	Type	Default	Description
position	string		Specifies the location to which items should be moved. Following values are possible: "default": Output location is determined by Service. "left": Present items to left side of device. "right": Present items to right side of device. "center": Present items to center output position. "top": Present items to the top output position. "bottom": Present items to the bottom output position. "front": Present items to the front output position. "rear": Present items to the rear output position. "reject": Reject bin is used as output location.
physicalPositionName	string		Specifies which physical cash unit to empty and count. This name is the same as the *physicalPositionName* in the CashManagement.CashUnitInfo completion message.

Example Message (generated)

```
{
  "headers": {
    "requestId": "b34800d0-9dd2-4d50-89ea-92d1b13df54b",
    "type": "command",
    "name": "string"
  },
  "payload": {
    "timeout": "5000",
    "emptyAll": true,
    "position": "default",
    "physicalPositionName": "string"
  }
}
```

Completion Message

Message Header

Name	Type	Default	Description
requestId (Required)	string		Unique request identifier supplied by the client used to correlate the command with responses, events and completions. For Unsolicited Events the field will be empty.
type (Required)	string		The message type, either command, response, event or completion.
name (Required)	string		The original message name, for example "CardReader.Status"

Message Payload

Name	Type	Default	Description
status	string		ok if the commmand was successful otherwise error
errorDescription	string		If error, identified that cause of the error
	object		
countedPhysCUs	array		Array of counted physical cash unit objects.
countedPhysCUs.physicalPositionName	string		Specifies which physical cash unit was emptied and counted. This name is the same as the *physicalPositionName* in the CashManagement.CashUnitInfo completion message.
countedPhysCUs.unitId	string		Cash unit ID. This is the identifier defined in the *unitID* field in the CashManagement.CashUnitInfo completion message.
countedPhysCUs.dispensed	integer		The number of items that were dispensed during the emptying of the cash unit.
countedPhysCUs.counted	integer		The number of items that were counted during the emptying of the cash unit.

Name	Type	Default	Description
countedPhysCUs.pStatus	string		Supplies the status of the physical cash unit. Following values are possible: "ok": The cash unit is in a good state. "full": The cash unit is full. "high": The cash unit is almost full (i.e. reached or exceeded the threshold defined by *maximum*). "low": The cash unit is almost empty (i.e. reached or below the threshold defined by *minimum*). "empty": The cash unit is empty, or insufficient items in the cash unit are preventing further dispense operations. "inoperative": The cash unit is inoperative. "missing": The cash unit is missing. "noValue": The values of the specified cash unit are not available. "noReference": There is no reference value available for the notes in this cash unit. The cash unit has not been calibrated. "manuelInsertion": The cash unit has been inserted (including removal followed by a reinsertion) when the device was not in the exchange state. This cash unit cannot be dispensed from.

Example Message (generated)

```
{
  "headers": {
    "requestId": "b34800d0-9dd2-4d50-89ea-92d1b13df54b",
    "type": "command",
    "name": "string"
  },
  "payload": {
    "status": "ok",
    "errorDescription": "string",
    "countedPhysCUs": [
      {
        "physicalPositionName": "string",
        "unitId": "string",
        "dispensed": 0,
        "counted": 0,
        "pStatus": "ok"
      }
    ]
  }
}
```

Event Messages

- CashManagement.CashUnitErrorEvent
- Dispenser.ItemsTakenEvent
- Dispenser.ItemsPresentedEvent
- CashManagement.NoteErrorEvent
- CashManagement.InputP6Event
- CashManagement.InfoAvailableEvent
- Dispenser.ShutterStatusChangedEvent

Dispenser.PrepareDispense

Description

On some hardware it can take a significant amount of time for the dispenser to get ready to dispense media. On this type of hardware the Dispenser.PrepareDispense command can be used to improve transaction performance.

If this command is supported (see the *prepareDispense* capability) then applications can help to improve the time taken to dispense media by issuing this command as soon as the application knows that a dispense is likely to happen. This command either prepares the device for the next dispense operation, or terminates the dispense preparation if the subsequent dispense operation is no longer required.

With the exception of the Dipenser.Denominate and Dispenser.Dispense commands, which will not stop the dispense preparation, any execute command on Dispenser or CashAcceptor will automatically stop the dispense preparation.

If this command is executed and the device is already in the specified *action* state, then this execution will have no effect and will complete with a successful completion message.

Command Message

Message Header

Name	Type	Default	Description

Name	Type	Default	Description
requestId (Required)	string		Unique request identifier supplied by the client used to correlate the command with responses, events and completions. For Unsolicited Events the field will be empty.
type (Required)	string		The message type, either command, response, event or completion.
name (Required)	string		The original message name, for example "CardReader.Status"

Message Payload

Name	Type	Default	Description
timeout	integer	0	Timeout in milliseconds for the command to complete. If set to zero, the command will not timeout but can be cancelled.
	object		
action	string		A value specifying the type of actions. Following values are possible: "start": Initiates the action to prepare for the next dispense command. This command does not wait until the device is ready to dispense before returning a completion event, it completes as soon as the preparation has been initiated. "stop": Stops the previously activated dispense preparation. For example the motor of the transport will be stopped. This should be used if for some reason the subsequent dispense operation is no longer required.

Example Message (generated)

```
{
  "headers": {
    "requestId": "b34800d0-9dd2-4d50-89ea-92d1b13df54b",
    "type": "command",
    "name": "string"
  },
  "payload": {
    "timeout": "5000",
    "action": "start"
  }
}
```

Completion Message

Message Header

Name	Type	Default	Description
requestId (Required)	string		Unique request identifier supplied by the client used to correlate the command with responses, events and completions. For Unsolicited Events the field will be empty.
type (Required)	string		The message type, either command, response, event or completion.
name (Required)	string		The original message name, for example "CardReader.Status"

Message Payload

Name	Type	Default	Description
status	string		ok if the commmand was successful otherwise error
errorDescription	string		If error, identified that cause of the error

Example Message (generated)

```
{
  "headers": {
    "requestId": "b34800d0-9dd2-4d50-89ea-92d1b13df54b",
    "type": "command",
    "name": "string"
  },
  "payload": {
    "status": "ok",
    "errorDescription": "string"
  }
}
```

Event Messages

Unsolicited Events

Dispenser.SafeDoorOpenEvent

Description

This event is generated when the safe door has been opened.

Message Header

Name	Type	Default	Description
requestId (Required)	string		Unique request identifier supplied by the client used to correlate the command with responses, events and completions. For Unsolicited Events the field will be empty.
type (Required)	string		The message type, either command, response, event or completion.
name (Required)	string		The original message name, for example "CardReader.Status"

Example Message (generated)

```
{
  "headers": {
    "requestId": "b34800d0-9dd2-4d50-89ea-92d1b13df54b",
    "type": "command",
    "name": "string"
  }
}
```

Dispenser.SafeDoorClosedEvent

Description

This event is generated when the safe door has been closed.

Message Header

Name	Type	Default	Description
requestId (Required)	string		Unique request identifier supplied by the client used to correlate the command with responses, events and completions. For Unsolicited Events the field will be empty.
type (Required)	string		The message type, either command, response, event or completion.
name (Required)	string		The original message name, for example "CardReader.Status"

Example Message (generated)

```
{
  "headers": {
    "requestId": "b34800d0-9dd2-4d50-89ea-92d1b13df54b",
    "type": "command",
    "name": "string"
  }
}
```

Dispenser.ItemsTakenEvent

Description

This event is generated when items presented to the user have been taken. This event may be generated at any time.

Message Header

Name	Type	Default	Description
requestId (Required)	string		Unique request identifier supplied by the client used to correlate the command with responses, events and completions. For Unsolicited Events the field will be empty.
type (Required)	string		The message type, either command, response, event or completion.
name (Required)	string		The original message name, for example "CardReader.Status"

Message Payload

Name	Type	Default	Description

Name	Type	Default	Description
position	string		The output position from which the items have been removed. Following values are possible: "default": The default configuration. "left": The left output position. "right": The right output position. "center": The center output position. "top": The top output position. "bottom": The bottom output position. "front": The front output position. "rear": The rear output position.

Example Message (generated)

```
{
  "headers": {
    "requestId": "b34800d0-9dd2-4d50-89ea-92d1b13df54b",
    "type": "command",
    "name": "string"
  },
  "payload": {
    "position": "default"
  }
}
```

Dispenser.ItemsPresentedEvent

Description

This event specifies that items have been presented to the user during a count operation and need to be taken.

Message Header

Name	Type	Default	Description

Name	Type	Default	Description
requestId (Required)	string		Unique request identifier supplied by the client used to correlate the command with responses, events and completions. For Unsolicited Events the field will be empty.
type (Required)	string		The message type, either command, response, event or completion.
name (Required)	string		The original message name, for example "CardReader.Status"

Example Message (generated)

```
{
  "headers": {
    "requestId": "b34800d0-9dd2-4d50-89ea-92d1b13df54b",
    "type": "command",
    "name": "string"
  }
}
```

Dispenser.MediaDetectedEvent

Description

This service event is generated if media is detected during a reset command. The payload on the event informs the application of the position of the media after the reset completes. If the device has been unable to successfully move the items found then this payload will be ommitted.

Message Header

Name	Type	Default	Description
requestId (Required)	string		Unique request identifier supplied by the client used to correlate the command with responses, events and completions. For Unsolicited Events the field will be empty.
type (Required)	string		The message type, either command, response, event or completion.
name (Required)	string		The original message name, for example "CardReader.Status"

Message Payload

Name	Type	Default	Description
number	integer		If non-zero, this value specifies the *number* (as specified by CashManagement.CashUnitInfo) of the single cash unit to be used for the storage of any items found.

Name	Type	Default	Description
retractArea	object		This field is used if items are to be moved to internal areas of the device, including cash units, the intermediate stacker, or the transport.
retractArea.outputPosition	string		Output position from which to retract the items. Following values are possible: "default": The default configuration information should be used. "left": Retract items from the left output position. "right": Retract items from the right output position. "center": Retract items from the center output position. "top": Retract items from the top output position. "bottom": Retract items from the bottom output position. "front": Retract items from the front output position. "rear": Retract items from the rear output position.
retractArea.retractArea	string		This value specifies the area to which the items are to be retracted. Following values are possible: "retract": Retract the items to a retract cash unit. "transport": Retract the items to the transport. "stacker": Retract the items to the intermediate stacker area. "reject": Retract the items to a reject cash unit. "itemCassette": Retract the items to the item cassettes, i.e. cassettes that can be dispensed from.

Name	Type	Default	Description
retractArea.index	integer		If *retractArea* is set to "retract" this field defines the position inside the retract cash units into which the cash is to be retracted. *index* starts with a value of one (1) for the first retract position and increments by one for each subsequent position. If there are several logical retract cash units (of type "retractCassette" in command CashManagement.CashUnitInfo), *index* would be incremented from the first position of the first retract cash unit to the last position of the last retract cash unit. The maximum value of *index* is the sum of *maximum* of each retract cash unit. If *retractArea* is not set to "retract" the value of this field is ignored.
outputPosition	string		The output position to which items are to be moved. This field is only used if *number* is zero and retractArea is omitted. Following values are possible: "default": The default configuration. "left": The left output position. "right": The right output position. "center": The center output position. "top": The top output position. "bottom": The bottom output position. "front": The front output position. "rear": The rear output position.

Example Message (generated)

```
{
  "headers": {
    "requestId": "b34800d0-9dd2-4d50-89ea-92d1b13df54b",
    "type": "command",
    "name": "string"
  },
  "payload": {
    "number": 0,
    "retractArea": {
      "outputPosition": "default",
      "retractArea": "retract",
      "index": 0
    },
    "outputPosition": "default"
  }
}
```

Dispenser.ShutterStatusChangedEvent

Description

Within the limitations of the hardware sensors this event is generated whenever the status of a shutter changes. The shutter status can change because of an explicit, implicit or manual operation depending on how the shutter is operated.

Message Header

Name	Type	Default	Description
requestId (Required)	string		Unique request identifier supplied by the client used to correlate the command with responses, events and completions. For Unsolicited Events the field will be empty.
type (Required)	string		The message type, either command, response, event or completion.
name (Required)	string		The original message name, for example "CardReader.Status"

Message Payload

Name	Type	Default	Description

Name	Type	Default	Description
position	string		Specifies one of the Dispenser output positions whose shutter status has changed. Following values are possible: "left": Left output position. "right": Right output position. "center": Center output position. "top": Top output position. "bottom": Bottom output position. "front": Front output position. "rear": Rear output position.
shutter	string		Specifies the new state of the shutter. Following values are possible: "closed": The shutter is closed. "open": The shutter is opened. "jammed": The shutter is jammed. "unknown": Due to a hardware error or other condition, the state of the shutter cannot be determined.

Example Message (generated)

```
{
  "headers": {
    "requestId": "b34800d0-9dd2-4d50-89ea-92d1b13df54b",
    "type": "command",
    "name": "string"
  },
  "payload": {
    "position": "left",
    "shutter": "closed"
  }
}
```

Events

CashManagement.CashUnitErrorEvent

Description

This event is generated if there is a problem with a cash unit during the execution of a command.

Message Header

Name	Type	Default	Description
requestId (Required)	string		Unique request identifier supplied by the client used to correlate the command with responses, events and completions. For Unsolicited Events the field will be empty.
type (Required)	string		The message type, either command, response, event or completion.
name (Required)	string		The original message name, for example "CardReader.Status"

Message Payload

Name	Type	Default	Description
failure	string		Specifies the kind of failure that occurred in the cash unit. Following values are possible: "empty": Specified cash unit is empty. "error": Specified cash unit has malfunctioned. "full": Specified cash unit is full. "locked": Specified cash unit is locked. "invalid": Specified cash unit is invalid. "config": An attempt has been made to change the settings of a self-configuring cash unit. "notConfigured": Specified cash unit is not configured.
cashUnit.	object		
cashUnit.number	integer		Index number of the cash unit structure. Each structure has a unique logical number starting with a value of one (1) for the first structure, and incrementing by one for each subsequent structure.

Name	Type	Default	Description
cashUnit.type	string		Type of cash unit. Following values are possible: "notApplicable": Not applicable. Typically means cash unit is missing. "rejectCassette": Reject cash unit. This type will also indicate a combined reject/retract cash unit. "billCassette": Cash unit containing bills. "coinCylinder": Coin cylinder. "coinDispenser": Coin dispenser as a whole unit. "retractCassette": Retract cash unit. "coupon": Cash unit containing coupons or advertising material. "document": Cash unit containing documents. "replenishmentContainer": Replenishment container. A cash unit can be refilled from a replenishment container. "recycling": Recycling cash unit. This unit is only present when the device implements the Dispenser and CashAcceptor interfaces. "cashIn": Cash-in cash unit.
cashUnit.unitID	string		The Cash Unit Identifier.
cashUnit.currencyID	string		A three character string storing the ISO format [Ref. 2] Currency ID. This value will be omitted for cash units which contain items of more than one currency type or items to which currency is not applicable. If the *status* field for this cash unit is *noValue* it is the responsibility of the application to assign a value to this field. This value is persistent.
cashUnit.values	integer		Supplies the value of a single item in the cash unit. This value is expressed in minimum dispense units (see command CashManagement.CurrencyExp). If the *currencyID* field for this cash unit is omitted, then this field will contain zero. If the *status* field for this cash unit is *noValue* it is the responsibility of the application to assign a value to this field. This value is persistent.

Name	Type	Default	Description
cashUnit.count	integer		The meaning of this count depends on the type of cash unit. This value is persistent. For all cash units except retract cash units (*type* is not *retractCassette*) this value specifies the number of items inside all the physical cash units associated with this cash unit. For all dispensing cash units (*type* is *billCassette*, *coinCylinder*, *coinDispenser*, *coupon*, *document* or *recycling*), this value includes any items from the physical cash units not yet presented to the customer. This count is only decremented when the items are either known to be in customer access or successfully rejected. If the cash unit is usable from the CashAcceptor interface (*type* is *recycling*, *cashIn*, *retractCassette* or *rejectCassette*) then this value will be incremented as a result of a cash-in operation. Note that for a reject cash unit (*type* is *rejectCassette*), this value is unreliable, since the typical reason for dumping items to the reject cash unit is a suspected count failure. For a retract cash unit (*type* is *retractCassette*) this value specifies the number of retract operations which result in items entering the cash unit.
cashUnit.maximum	integer		This field is only applicable to retract and reject cash units. When ulCount reaches this value the threshold event CashManagement.CashUnitThresholdEvent (*high*) will be generated. If this value is non-zero then hardware sensors in the device do not trigger threshold events. If this value is zero then hardware sensors will trigger threshold events if *hardwareSensor* is TRUE. This value is persistent.

Name	Type	Default	Description
cashUnit.status	string		Supplies the status of the cash unit. Following values are possible: "ok": The cash unit is in a good state. "full": The cash unit is full. "high": The cash unit is almost full (i.e. reached or exceeded the threshold defined by *maximum*). "low": The cash unit is almost empty (i.e. reached or below the threshold defined by *minimum*). "empty": The cash unit is empty, or insufficient items in the cash unit are preventing further dispense operations. "inoperative": The cash unit is inoperative. "missing": The cash unit is missing. "noValue": The values of the specified cash unit are not available. "noReference": There is no reference value available for the notes in this cash unit. The cash unit has not been calibrated. "manuelInsertion": The cash unit has been inserted (including removal followed by a reinsertion) when the device was not in the exchange state. This cash unit cannot be dispensed from.
cashUnit.appLock	boolean		If this value is TRUE items cannot be dispensed from or deposited into the cash unit. If this value is TRUE and the application attempts to use the cash unit a CashManagement.CashUnitErrorEvent event will be generated and an error completion message will be returned. This value is persistent.
cashUnit.physical	array		Array of pyhiscal cash unit objects.

Name	Type	Default	Description
cashUnit.cashUnitName	string		A name which helps to identify the logical type of the cash unit. This is especially useful in the case of cash units of type *document* where different documents can have the same currency and value. For example, travelers checks and bank checks may have the same currency and value but still need to be identifiable as different types of document. Where this value is not relevant (e.g. in bill cash units) the property can be omitted. This value is persistent.
cashUnit.initialCount	integer		Initial number of items contained in the cash unit. This value is persistent.
cashUnit.dispensedCount	integer		The number of items dispensed from all the physical cash units associated with this cash unit. This count is incremented when the items are removed from any of the associated physical cash units. This count includes any items that were rejected during the dispense operation and are no longer in this cash unit. This field is always zero for cash units with a *type* of *rejectCassette* or *retractCassette*. This value is persistent.
cashUnit.presentedCount	integer		The number of items dispensed from all the physical cash units associated with this cash unit. This count is incremented when the items are removed from any of the associated physical cash units. This count includes any items that were rejected during the dispense operation and are no longer in this cash unit. This field is always zero for cash units with a *type* of *rejectCassette* or *retractCassette*. This value is persistent.
cashUnit.retractedCount	integer		The number of items that have been accessible to a customer and retracted into all the physical cash units associated with this cash unit. This value is persistent.

Name	Type	Default	Description
cashUnit.rejectCount	integer		The number of items dispensed from this cash unit which have been rejected, are in a cash unit other than this cash unit, and which have not been accessible to a customer. This value may be unreliable, since a typical reason for rejecting items is a suspected pick failure. Other reasons for rejecting items may include incorrect note denominations, classifications not valid for dispensing, or where the transaction has been cancelled and a Reject command has been called. For reject and retract cash units (*type* is *rejectCassette* or *retractCassette*) this field does not apply and will be reported as zero. This value is persistent.
cashUnit.minimum	integer		This field is not applicable to retract and reject cash units. For all other cash units, when *count* reaches this value the threshold event CashManagement.CashUnitThresholdEvent (*low*) will be generated. If this value is non-zero then hardware sensors in the device do not trigger threshold events. If this value is zero then hardware sensors will trigger threshold events if *hardwareSensor* is TRUE. This value is persistent.
cashUnit.	object		
cashUnit.itemType	object		Specifies the type of items the cash unit takes as a combination of the following flags. The table in the Comments section of this command defines how to interpret the combination of these flags (TODO: include Table)
cashUnit.itemType.all	boolean		The cash unit takes all fit banknote types. These are level 4 notes which are fit for recycling.
cashUnit.itemType.unfit	boolean		The cash unit takes all unfit banknotes. These are level 4 notes which are unfit for recycling.
cashUnit.itemType.individual	boolean		The cash unit takes all types of fit banknotes specified in an individual list. These are level 4 notes which are fit for recycling.
cashUnit.itemType.level1	boolean		Level 1 note types are stored in this cash unit.

Name	Type	Default	Description
cashUnit.itemType.level2	boolean		If notes can be classified as level 2, then level 2 note types are stored in this cash unit.
cashUnit.itemType.level3	boolean		If notes can be classified as level 3, then level 3 note types are stored in this cash unit.
cashUnit.itemType.itemProcessor	boolean		The cash unit can accept items on the ItemProcessor interface.
cashUnit.itemType.unfitIndividual	boolean		The cash unit takes all types of unfit banknotes specified in an individual list. These are level 4 notes which are unfit for recycling.
cashUnit.cashInCount	integer		Count of items that have entered the logical cash unit. This counter is incremented whenever an item enters a physical cash unit that belongs to this logical cash unit for any reason, unless it originated from this cash unit but was returned without being accessible to a customer. For a retract cash unit this value represents the total number of items of all types in the cash unit, or if the device cannot count items during a retract operation this value will be zero. This value is persistent.
cashUnit.noteNumberList	object		Array of cash items inside the cash unit. The content of this structure is persistent. If the cash unit is Dispenser specific cash unit with *type billCassette* or the contents of the cash unit are not known this structure will be omitted. If the cash unit is of *type retractCassette* this pointer will be omitted except for the following cases: • If the retract cash unit is configured to accept level 2 notes then the number and type of level 2 notes is returned in the *noteNumberList* and *count* contains the number of retract operations. *cashInCount* contains the actual number of level 2 notes. • If items are recognized during retract operations then the number and type of notes retracted is returned in *noteNumberList* and *count* contains the number of retract operations. *cashInCount* contains the actual number of retracted items.

Name	Type	Default	Description
cashUnit.noteNumberList.noteNumber	array		Array of banknote numbers the cash unit contains.
cashUnit.noteNumberList.noteNumber.noteID	integer		Identification of note type. The Note ID represents the note identifiers reported by the *CashAcceptor.BanknoteTypes* command. If this value is zero then the note type is unknown.
cashUnit.noteNumberList.noteNumber.count	integer		Actual count of cash items. The value is incremented each time cash items are moved to a cash unit. In the case of recycle cash units this count is decremented as defined in the description of the logical *count* field.
cashUnit.noteIDs	array		Array of integers which contains the note IDs of the banknotes the cash-in cash unit or recycle cash unit can take. This field only applies to *individual* cassette types. If there are no note IDs defined for the cassette or the cassette is not defined as *individual* then *noteIDs* will be omitted.
cashUnit.extra	array		Pointer to a list of vendor-specific information about the logical cash unit. The information is returned as a series of “key=value” strings so that it is easily extensible by Service Providers.

Example Message (generated)

```
{
  "headers": {
    "requestId": "b34800d0-9dd2-4d50-89ea-92d1b13df54b",
    "type": "command",
    "name": "string"
  },
  "payload": {
    "failure": "empty",
    "cashUnit": {
      "number": 0,
      "type": "notApplicable",
      "unitID": "string",
      "currencyID": "string",
      "values": 0,
      "count": 0,
      "maximum": 0,
      "status": "ok",
      "appLock": true,
      "physical": [
        {
          "physicalPositionName": "string",
          "unitID": "string",
          "count": 0,
          "maximum": 0,
          "pStatus": "ok",
          "hardwareSensor": true,
          "initialCount": 0,
          "dispensedCount": 0,
          "presentedCount": 0,
          "retractedCount": 0,
          "rejectCount": 0,
          "cashInCount": 0,
          "extra": [
            "string"
          ]
        }
      ],
      "cashUnitName": "string",
      "initialCount": 0,
      "dispensedCount": 0,
      "presentedCount": 0,
      "retractedCount": 0,
      "rejectCount": 0,
      "minimum": 0,
      "itemType": {
        "all": true,
        "unfit": true,
        "individual": true,
        "level1": true,
        "level2": true,
        "level3": true,
        "itemProcessor": true,
        "unfitIndividual": true
      },
      "cashInCount": 0,
      "noteNumberList": {
        "noteNumber": [
          {
            "noteID": 0,
            "count": 0
          }
        ]
      },
      "noteIDs": [
        0
      ],
      "extra": [
        "string"
      ]
    }
  }
}
```

CashManagement.CashUnitThresholdEvent

Description

This user event is generated when a threshold condition has occurred in one of the logical cash units. This event can be triggered either by hardware sensors in the device or by the logical *count* reaching the *minimum* or *maximum* value as specified in the CashUnitInfo structure. The application can check if the device has hardware sensors by querying the *hardwareSensor* field of the physical cash unit structure. If any of the physical cash units associated with the logical cash unit have this capability then threshold events based on hardware sensors will be triggered if the *maximum* or *minimum* values are not used and are set to zero. In the situation where the cash unit is associated with multiple physical cash units the CashManagement.CashUnitInfoChangedEvent will be generated when any of the physical cash units reaches the threshold. When the final physical cash unit reaches the threshold, the CashManagement.CashUnitThresholdEvent as well as the CashManagement.CashUnitInfoChangedEvent event will be generated.

Message Header

Name	Type	Default	Description
requestId (Required)	string		Unique request identifier supplied by the client used to correlate the command with responses, events and completions. For Unsolicited Events the field will be empty.
type (Required)	string		The message type, either command, response, event or completion.
name (Required)	string		The original message name, for example "CardReader.Status"

Message Payload

Name	Type	Default	Description
	object		
number	integer		Index number of the cash unit structure. Each structure has a unique logical number starting with a value of one (1) for the first structure, and incrementing by one for each subsequent structure.

Name	Type	Default	Description
type	string		Type of cash unit. Following values are possible: "notApplicable": Not applicable. Typically means cash unit is missing. "rejectCassette": Reject cash unit. This type will also indicate a combined reject/retract cash unit. "billCassette": Cash unit containing bills. "coinCylinder": Coin cylinder. "coinDispenser": Coin dispenser as a whole unit. "retractCassette": Retract cash unit. "coupon": Cash unit containing coupons or advertising material. "document": Cash unit containing documents. "replenishmentContainer": Replenishment container. A cash unit can be refilled from a replenishment container. "recycling": Recycling cash unit. This unit is only present when the device implements the Dispenser and CashAcceptor interfaces. "cashIn": Cash-in cash unit.
unitID	string		The Cash Unit Identifier.
currencyID	string		A three character string storing the ISO format [Ref. 2] Currency ID. This value will be omitted for cash units which contain items of more than one currency type or items to which currency is not applicable. If the *status* field for this cash unit is *noValue* it is the responsibility of the application to assign a value to this field. This value is persistent.
values	integer		Supplies the value of a single item in the cash unit. This value is expressed in minimum dispense units (see command CashManagement.CurrencyExp). If the *currencyID* field for this cash unit is omitted, then this field will contain zero. If the *status* field for this cash unit is *noValue* it is the responsibility of the application to assign a value to this field. This value is persistent.

Name	Type	Default	Description
count	integer		The meaning of this count depends on the type of cash unit. This value is persistent. For all cash units except retract cash units (*type* is not *retractCassette*) this value specifies the number of items inside all the physical cash units associated with this cash unit. For all dispensing cash units (*type* is *billCassette*, *coinCylinder*, *coinDispenser*, *coupon*, *document* or *recycling*), this value includes any items from the physical cash units not yet presented to the customer. This count is only decremented when the items are either known to be in customer access or successfully rejected. If the cash unit is usable from the CashAcceptor interface (*type* is *recycling*, *cashIn*, *retractCassette* or *rejectCassette*) then this value will be incremented as a result of a cash-in operation. Note that for a reject cash unit (*type* is *rejectCassette*), this value is unreliable, since the typical reason for dumping items to the reject cash unit is a suspected count failure. For a retract cash unit (*type* is *retractCassette*) this value specifies the number of retract operations which result in items entering the cash unit.
maximum	integer		This field is only applicable to retract and reject cash units. When ulCount reaches this value the threshold event CashManagement.CashUnitThresholdEvent (*high*) will be generated. If this value is non-zero then hardware sensors in the device do not trigger threshold events. If this value is zero then hardware sensors will trigger threshold events if *hardwareSensor* is TRUE. This value is persistent.

Name	Type	Default	Description
status	string		Supplies the status of the cash unit. Following values are possible: "ok": The cash unit is in a good state. "full": The cash unit is full. "high": The cash unit is almost full (i.e. reached or exceeded the threshold defined by *maximum*). "low": The cash unit is almost empty (i.e. reached or below the threshold defined by *minimum*). "empty": The cash unit is empty, or insufficient items in the cash unit are preventing further dispense operations. "inoperative": The cash unit is inoperative. "missing": The cash unit is missing. "noValue": The values of the specified cash unit are not available. "noReference": There is no reference value available for the notes in this cash unit. The cash unit has not been calibrated. "manuelInsertion": The cash unit has been inserted (including removal followed by a reinsertion) when the device was not in the exchange state. This cash unit cannot be dispensed from.
appLock	boolean		If this value is TRUE items cannot be dispensed from or deposited into the cash unit. If this value is TRUE and the application attempts to use the cash unit a CashManagement.CashUnitErrorEvent event will be generated and an error completion message will be returned. This value is persistent.
physical	array		Array of pyhiscal cash unit objects.

Name	Type	Default	Description
cashUnitName	string		A name which helps to identify the logical type of the cash unit. This is especially useful in the case of cash units of type *document* where different documents can have the same currency and value. For example, travelers checks and bank checks may have the same currency and value but still need to be identifiable as different types of document. Where this value is not relevant (e.g. in bill cash units) the property can be omitted. This value is persistent.
initialCount	integer		Initial number of items contained in the cash unit. This value is persistent.
dispensedCount	integer		The number of items dispensed from all the physical cash units associated with this cash unit. This count is incremented when the items are removed from any of the associated physical cash units. This count includes any items that were rejected during the dispense operation and are no longer in this cash unit. This field is always zero for cash units with a *type* of *rejectCassette* or *retractCassette*. This value is persistent.
presentedCount	integer		The number of items dispensed from all the physical cash units associated with this cash unit. This count is incremented when the items are removed from any of the associated physical cash units. This count includes any items that were rejected during the dispense operation and are no longer in this cash unit. This field is always zero for cash units with a *type* of *rejectCassette* or *retractCassette*. This value is persistent.
retractedCount	integer		The number of items that have been accessible to a customer and retracted into all the physical cash units associated with this cash unit. This value is persistent.

Name	Type	Default	Description
rejectCount	integer		The number of items dispensed from this cash unit which have been rejected, are in a cash unit other than this cash unit, and which have not been accessible to a customer. This value may be unreliable, since a typical reason for rejecting items is a suspected pick failure. Other reasons for rejecting items may include incorrect note denominations, classifications not valid for dispensing, or where the transaction has been cancelled and a Reject command has been called. For reject and retract cash units (*type* is *rejectCassette* or *retractCassette*) this field does not apply and will be reported as zero. This value is persistent.
minimum	integer		This field is not applicable to retract and reject cash units. For all other cash units, when *count* reaches this value the threshold event CashManagement.CashUnitThresholdEvent (*low*) will be generated. If this value is non-zero then hardware sensors in the device do not trigger threshold events. If this value is zero then hardware sensors will trigger threshold events if *hardwareSensor* is TRUE. This value is persistent.
	object		
itemType	object		Specifies the type of items the cash unit takes as a combination of the following flags. The table in the Comments section of this command defines how to interpret the combination of these flags (TODO: include Table)
itemType.all	boolean		The cash unit takes all fit banknote types. These are level 4 notes which are fit for recycling.
itemType.unfit	boolean		The cash unit takes all unfit banknotes. These are level 4 notes which are unfit for recycling.
itemType.individual	boolean		The cash unit takes all types of fit banknotes specified in an individual list. These are level 4 notes which are fit for recycling.
itemType.level1	boolean		Level 1 note types are stored in this cash unit.

Name	Type	Default	Description
itemType.level2	boolean		If notes can be classified as level 2, then level 2 note types are stored in this cash unit.
itemType.level3	boolean		If notes can be classified as level 3, then level 3 note types are stored in this cash unit.
itemType.itemProcessor	boolean		The cash unit can accept items on the ItemProcessor interface.
itemType.unfitIndividual	boolean		The cash unit takes all types of unfit banknotes specified in an individual list. These are level 4 notes which are unfit for recycling.
cashInCount	integer		Count of items that have entered the logical cash unit. This counter is incremented whenever an item enters a physical cash unit that belongs to this logical cash unit for any reason, unless it originated from this cash unit but was returned without being accessible to a customer. For a retract cash unit this value represents the total number of items of all types in the cash unit, or if the device cannot count items during a retract operation this value will be zero. This value is persistent.
noteNumberList	object		Array of cash items inside the cash unit. The content of this structure is persistent. If the cash unit is Dispenser specific cash unit with *type billCassette* or the contents of the cash unit are not known this structure will be omitted. If the cash unit is of *type retractCassette* this pointer will be omitted except for the following cases: • If the retract cash unit is configured to accept level 2 notes then the number and type of level 2 notes is returned in the *noteNumberList* and *count* contains the number of retract operations. *cashInCount* contains the actual number of level 2 notes. • If items are recognized during retract operations then the number and type of notes retracted is returned in *noteNumberList* and *count* contains the number of retract operations. *cashInCount* contains the actual number of retracted items.

Name	Type	Default	Description
noteNumberList.noteNumber	array		Array of banknote numbers the cash unit contains.
noteNumberList.noteNumber.noteID	integer		Identification of note type. The Note ID represents the note identifiers reported by the *CashAcceptor.BanknoteTypes* command. If this value is zero then the note type is unknown.
noteNumberList.noteNumber.count	integer		Actual count of cash items. The value is incremented each time cash items are moved to a cash unit. In the case of recycle cash units this count is decremented as defined in the description of the logical *count* field.
noteIDs	array		Array of integers which contains the note IDs of the banknotes the cash-in cash unit or recycle cash unit can take. This field only applies to *individual* cassette types. If there are no note IDs defined for the cassette or the cassette is not defined as *individual* then *noteIDs* will be omitted.
extra	array		Pointer to a list of vendor-specific information about the logical cash unit. The information is returned as a series of “key=value” strings so that it is easily extensible by Service Providers.

Example Message (generated)

```
{
  "headers": {
    "requestId": "b34800d0-9dd2-4d50-89ea-92d1b13df54b",
    "type": "command",
    "name": "string"
  },
  "payload": {
    "number": 0,
    "type": "notApplicable",
    "unitID": "string",
    "currencyID": "string",
    "values": 0,
    "count": 0,
    "maximum": 0,
    "status": "ok",
    "appLock": true,
    "physical": [
      {
        "physicalPositionName": "string",
        "unitID": "string",
        "count": 0,
        "maximum": 0,
        "pStatus": "ok",
        "hardwareSensor": true,
        "initialCount": 0,
        "dispensedCount": 0,
        "presentedCount": 0,
        "retractedCount": 0,
        "rejectCount": 0,
        "cashInCount": 0,
        "extra": [
          "string"
        ]
      }
    ],
    "cashUnitName": "string",
    "initialCount": 0,
    "dispensedCount": 0,
    "presentedCount": 0,
    "retractedCount": 0,
    "rejectCount": 0,
    "minimum": 0,
    "itemType": {
      "all": true,
      "unfit": true,
      "individual": true,
      "level1": true,
      "level2": true,
      "level3": true,
      "itemProcessor": true,
      "unfitIndividual": true
    },
    "cashInCount": 0,
    "noteNumberList": {
      "noteNumber": [
        {
          "noteID": 0,
          "count": 0
        }
      ]
    },
    "noteIDs": [
      0
    ],
    "extra": [
      "string"
    ]
  }
}
```

Dispenser.DelayedDispenseEvent

Description

This event is generated if the start of a dispense operation has been delayed.

Message Header

Name	Type	Default	Description
requestId (Required)	string		Unique request identifier supplied by the client used to correlate the command with responses, events and completions. For Unsolicited Events the field will be empty.
type (Required)	string		The message type, either command, response, event or completion.
name (Required)	string		The original message name, for example "CardReader.Status"

Message Payload

Name	Type	Default	Description
delay	integer		The time in milliseconds by which the dispense operation will be delayed.

Example Message (generated)

```
{
  "headers": {
    "requestId": "b34800d0-9dd2-4d50-89ea-92d1b13df54b",
    "type": "command",
    "name": "string"
  },
  "payload": {
    "delay": 0
  }
}
```

Dispenser.StartDispenseEvent

Description

This event is generated when a delayed dispense operation begins.

Message Header

Name	Type	Default	Description
requestId (Required)	string		Unique request identifier supplied by the client used to correlate the command with responses, events and completions. For Unsolicited Events the field will be empty.
type (Required)	string		The message type, either command, response, event or completion.
name (Required)	string		The original message name, for example "CardReader.Status"

Message Payload

Name	Type	Default	Description

Name	Type	Default	Description
reqID	string		The requestId of the original dispense command.

Example Message (generated)

```
{
  "headers": {
    "requestId": "b34800d0-9dd2-4d50-89ea-92d1b13df54b",
    "type": "command",
    "name": "string"
  },
  "payload": {
    "reqID": "string"
  }
}
```

Dispenser.PartialDispenseEvent

Description

This event is generated when a dispense operation is divided into several sub-dispense operations because the hardware capacity of the Dispenser is exceeded.

Message Header

Name	Type	Default	Description
requestId (Required)	string		Unique request identifier supplied by the client used to correlate the command with responses, events and completions. For Unsolicited Events the field will be empty.
type (Required)	string		The message type, either command, response, event or completion.
name (Required)	string		The original message name, for example "CardReader.Status"

Message Payload

Name	Type	Default	Description
dispNum	integer		The number of sub-dispense operations into which the dispense operation has been divided.

Example Message (generated)

```
{
  "headers": {
    "requestId": "b34800d0-9dd2-4d50-89ea-92d1b13df54b",
    "type": "command",
    "name": "string"
  },
  "payload": {
    "dispNum": 0
  }
}
```

Dispenser.SubDispenseOkEvent

Description

This event is generated when one of the sub-dispense operations into which the dispense operation was divided has finished successfully. Note that in this case the values in the payload structure report the amount and number of each denomination dispensed in the sub-dispense operation.

Message Header

Name	Type	Default	Description
requestId (Required)	string		Unique request identifier supplied by the client used to correlate the command with responses, events and completions. For Unsolicited Events the field will be empty.
type (Required)	string		The message type, either command, response, event or completion.
name (Required)	string		The original message name, for example "CardReader.Status"

Message Payload

Name	Type	Default	Description
currencyID	string		Identification of currency in ISO format [Ref. 2]. Where the denomination contains multiple currencies this field should be omitted.
amount	integer		The amount to be denominated or dispensed. Where the denomination contains multiple currencies this value is zero.
values	array		This list specifies the number of items to take from each of the cash units. This list corresponds to the array of cash unit structures returned by the last CashManagement.CashUnitInfo command or set by the last CashManagement.SetCashUnitInfo or CashManagement.EndExchange commands. The first value in the array is related to the cash structure with the index number 1. This array contains a field for each possible cash unit. If a cash unit is not required in the denomination its corresponding field in this array should be set to zero. If the application does not wish to specify a denomination, it should omit the values property.
cashBox	integer		Only applies to Teller Dispensers. Amount to be paid from the teller’s cash box.

Example Message (generated)

```
{
  "headers": {
    "requestId": "b34800d0-9dd2-4d50-89ea-92d1b13df54b",
    "type": "command",
    "name": "string"
  },
  "payload": {
    "currencyID": "string",
    "amount": 0,
    "values": [
      0
    ],
    "cashBox": 0
  }
}
```

Dispenser.IncompleteDispenseEvent

Description

This event is generated during Dispenser.Dispense when it has not been possible to dispense the entire denomination but part of the requested denomination is on the intermediate stacker or in customer access. Note that in this case the values in this structure report the amount and number of each denomination that are in customer access or on the intermediate stacker. Dispenser.PresentStatus can be used to determine whether the items are in customer access.

Message Header

Name	Type	Default	Description
requestId (Required)	string		Unique request identifier supplied by the client used to correlate the command with responses, events and completions. For Unsolicited Events the field will be empty.
type (Required)	string		The message type, either command, response, event or completion.
name (Required)	string		The original message name, for example "CardReader.Status"

Message Payload

Name	Type	Default	Description
currencyID	string		Identification of currency in ISO format [Ref. 2]. Where the denomination contains multiple currencies this field should be omitted.
amount	integer		The amount to be denominated or dispensed. Where the denomination contains multiple currencies this value is zero.

Name	Type	Default	Description
values	array		This list specifies the number of items to take from each of the cash units. This list corresponds to the array of cash unit structures returned by the last CashManagement.CashUnitInfo command or set by the last CashManagement.SetCashUnitInfo or CashManagement.EndExchange commands. The first value in the array is related to the cash structure with the index number 1. This array contains a field for each possible cash unit. If a cash unit is not required in the denomination its corresponding field in this array should be set to zero. If the application does not wish to specify a denomination, it should omit the values property.
cashBox	integer		Only applies to Teller Dispensers. Amount to be paid from the teller’s cash box.

Example Message (generated)

```
{
  "headers": {
    "requestId": "b34800d0-9dd2-4d50-89ea-92d1b13df54b",
    "type": "command",
    "name": "string"
  },
  "payload": {
    "currencyID": "string",
    "amount": 0,
    "values": [
      0
    ],
    "cashBox": 0
  }
}
```

CashManagement.NoteErrorEvent

Description

This event specifies the reason for a note detection error during the execution of a command.

Message Header

Name	Type	Default	Description
requestId (Required)	string		Unique request identifier supplied by the client used to correlate the command with responses, events and completions. For Unsolicited Events the field will be empty.
type (Required)	string		The message type, either command, response, event or completion.
name (Required)	string		The original message name, for example "CardReader.Status"

Message Payload

Name	Type	Default	Description

Name	Type	Default	Description
reason	string		The reason for the notes detection error. Following values are possible: "doubleNote": Double notes have been detected. "longNote": A long note has been detected. "skewedNote": A skewed note has been detected. "incorrectCount": An item counting error has occurred. "notesTooClose": Notes have been detected as being too close. "otherNoteError": An item error not covered by the other values has been detected. "shortNote": Short notes have been detected.

Example Message (generated)

```
{
  "headers": {
    "requestId": "b34800d0-9dd2-4d50-89ea-92d1b13df54b",
    "type": "command",
    "name": "string"
  },
  "payload": {
    "reason": "doubleNote"
  }
}
```

CashManagement.InputP6Event

Description

This event is generated if level 2 and / or level 3 notes are detected during the cash processing operation.

Message Header

Name	Type	Default	Description

Name	Type	Default	Description
requestId (Required)	string		Unique request identifier supplied by the client used to correlate the command with responses, events and completions. For Unsolicited Events the field will be empty.
type (Required)	string		The message type, either command, response, event or completion.
name (Required)	string		The original message name, for example "CardReader.Status"

Message Payload

Name	Type	Default	Description
p6Info	array		P& Info objects, one object for every level.
p6Info.level	string		Defines the note level. Following values are possible: "level2": Information for level 2 notes. "level3": Information for level 3 notes.
p6Info.noteNumberList	object		List of banknote types that were recognized as level 2 or level 3 notes.
p6Info.noteNumberList.noteNumber	array		Array of banknote numbers the cash unit contains.
p6Info.noteNumberList.noteNumber.noteID	integer		Identification of note type. The Note ID represents the note identifiers reported by the *CashAcceptor.BanknoteTypes* command. If this value is zero then the note type is unknown.
p6Info.noteNumberList.noteNumber.count	integer		Actual count of cash items. The value is incremented each time cash items are moved to a cash unit. In the case of recycle cash units this count is decremented as defined in the description of the logical *count* field.
p6Info.numOfSignatures	integer		Number of level 2 or level 3 signatures of this cash-in transaction. If it is zero no signatures are available.

Example Message (generated)

```
{
  "headers": {
    "requestId": "b34800d0-9dd2-4d50-89ea-92d1b13df54b",
    "type": "command",
    "name": "string"
  },
  "payload": {
    "p6Info": [
      {
        "level": "level2",
        "noteNumberList": {
          "noteNumber": [
            {
              "noteID": 0,
              "count": 0
            }
          ]
        },
        "numOfSignatures": 0
      }
    ]
  }
}
```

CashManagement.InfoAvailableEvent

Description

This execute event is generated when information is available for items detected during the cash processing operation.

Message Header

Name	Type	Default	Description
requestId (Required)	string		Unique request identifier supplied by the client used to correlate the command with responses, events and completions. For Unsolicited Events the field will be empty.
type (Required)	string		The message type, either command, response, event or completion.
name (Required)	string		The original message name, for example "CardReader.Status"

Message Payload

Name	Type	Default	Description
itemInfoSummary	array		Array of itemInfoSummary objects, one object for every level.

Name	Type	Default	Description
itemInfoSummary.level	string		Defines the note level. Following values are possible: "level1": Information for level 1 notes. "level2": Information for level 2 notes. "level3": Information for level 3 notes. "level4": Information for level 4 notes.
itemInfoSummary.numOfItems	integer		Number of items classified as *level* which have information available.

Example Message (generated)

```
{
  "headers": {
    "requestId": "b34800d0-9dd2-4d50-89ea-92d1b13df54b",
    "type": "command",
    "name": "string"
  },
  "payload": {
    "itemInfoSummary": [
      {
        "level": "level1",
        "numOfItems": 0
      }
    ]
  }
}
```

Dispenser.IncompleteRetractEvent

Description

This event is sent when a retract or reset command has completed with an error and not all of the items have been retracted.

Message Header

Name	Type	Default	Description
requestId (Required)	string		Unique request identifier supplied by the client used to correlate the command with responses, events and completions. For Unsolicited Events the field will be empty.

Name	Type	Default	Description
type (Required)	string		The message type, either command, response, event or completion.
name (Required)	string		The original message name, for example "CardReader.Status"

Message Payload

Name	Type	Default	Description
itemNumberList	object		The values in this structure report the amount and number of each denomination that were successfully moved during the command prior to the failure.
itemNumberList.itemNumber	array		Array of item number objects.
itemNumberList.itemNumber.currencyID	string		A three character array storing the ISO format [Ref. 2] Currency ID; if the currency of the item is not known this is omitted.
itemNumberList.itemNumber.values	integer		The value of a single item expressed in minimum dispense units; or a zero value if the value of the item is not known.
itemNumberList.itemNumber.release	integer		The release of the item. The higher this number is, the newer the release. Zero means that there is only one release or the release is not known. This value has not been standardized and therefore a release number of the same item will not necessarily have the same value in different systems.
itemNumberList.itemNumber.count	integer		The count of items of the same type moved to the same destination during the execution of this command.

Name	Type	Default	Description
itemNumberList.itemNumber.number	integer		The logical number of the cash unit which received items during the execution of this command. This value will be zero if items were moved to the *retractArea* "transport" or "stacker".
reason	string		The reason for not having retracted items. Following values are possible: "retractFailure": The retract has partially failed for a reason not covered by the other reasons listed in this event, for example failing to pick an item to be retracted. "retractAreaFull": The specified retract area (see input parameter *retractArea*) has become full during the retract operation. "foreignItemsDetected": Foreign items have been detected. "invalidBunch": An invalid bunch of items has been detected, e.g. it is too large or could not be processed.

Example Message (generated)

```
{
  "headers": {
    "requestId": "b34800d0-9dd2-4d50-89ea-92d1b13df54b",
    "type": "command",
    "name": "string"
  },
  "payload": {
    "itemNumberList": {
      "itemNumber": [
        {
          "currencyID": "string",
          "values": 0,
          "release": 0,
          "count": 0,
          "number": 0
        }
      ]
    },
    "reason": "retractFailure"
  }
}
```

CashManagement.CashUnitInfoChangedEvent

Description

This service event is generated under the following circumstances:

- It is generated whenever *status* and/or *pStatus* changes. For instance, a physical cash unit has been removed or inserted, or a physical/logical cash unit has become empty or full.
- This event will also be generated for every cash unit changed in any way (including changes to counts, e.g. *count, rejectCount, initialCount, dispensedCount* and *presentedCount*) as a result of the following commands:

  CashManagement.SetCashUnitInfo > CashManagement.EndExchange

- This event will also be fired when any change is made to a cash unit by the following commands, except for changes to counts (e.g. *count, rejectCount, initialCount, dispensedCount* and *presentedCount*):

  Dispenser.CalibrateCashUnit > Dispenser.TestCashUnit

- In addition this event will be generated when a cash unit has been counted during the CashAcceptor.CashUnitCount command execution. When a physical cash unit is removed, the status of the physical cash unit becomes "missing". If there are no physical cash units of the same logical type remaining the status of the logical type becomes "missing". When a physical cash unit is inserted and this physical cash unit is of an existing logical type, both the logical and the physical cash unit structures will be updated. If a physical cash unit of a new logical type is inserted the cash unit structure reported by the last CashManagement.CashUnitInfo command is no longer valid. In that case an application should issue a CashManagement.CashUnitInfo command after receiving this event to obtain updated cash unit information.

Message Header

Name	Type	Default	Description

Name	Type	Default	Description
requestId (Required)	string		Unique request identifier supplied by the client used to correlate the command with responses, events and completions. For Unsolicited Events the field will be empty.
type (Required)	string		The message type, either command, response, event or completion.
name (Required)	string		The original message name, for example "CardReader.Status"

Message Payload

Name	Type	Default	Description
	object		
number	integer		Index number of the cash unit structure. Each structure has a unique logical number starting with a value of one (1) for the first structure, and incrementing by one for each subsequent structure.
type	string		Type of cash unit. Following values are possible: "notApplicable": Not applicable. Typically means cash unit is missing. "rejectCassette": Reject cash unit. This type will also indicate a combined reject/retract cash unit. "billCassette": Cash unit containing bills. "coinCylinder": Coin cylinder. "coinDispenser": Coin dispenser as a whole unit. "retractCassette": Retract cash unit. "coupon": Cash unit containing coupons or advertising material. "document": Cash unit containing documents. "replenishmentContainer": Replenishment container. A cash unit can be refilled from a replenishment container. "recycling": Recycling cash unit. This unit is only present when the device implements the Dispenser and CashAcceptor interfaces. "cashIn": Cash-in cash unit.
unitID	string		The Cash Unit Identifier.

Name	Type	Default	Description
currencyID	string		A three character string storing the ISO format [Ref. 2] Currency ID. This value will be omitted for cash units which contain items of more than one currency type or items to which currency is not applicable. If the *status* field for this cash unit is *noValue* it is the responsibility of the application to assign a value to this field. This value is persistent.
values	integer		Supplies the value of a single item in the cash unit. This value is expressed in minimum dispense units (see command CashManagement.CurrencyExp). If the *currencyID* field for this cash unit is omitted, then this field will contain zero. If the *status* field for this cash unit is *noValue* it is the responsibility of the application to assign a value to this field. This value is persistent.
count	integer		The meaning of this count depends on the type of cash unit. This value is persistent. For all cash units except retract cash units (*type* is not *retractCassette*) this value specifies the number of items inside all the physical cash units associated with this cash unit. For all dispensing cash units (*type* is *billCassette*, *coinCylinder*, *coinDispenser*, *coupon*, *document* or *recycling*), this value includes any items from the physical cash units not yet presented to the customer. This count is only decremented when the items are either known to be in customer access or successfully rejected. If the cash unit is usable from the CashAcceptor interface (*type* is *recycling*, *cashIn*, *retractCassette* or *rejectCassette*) then this value will be incremented as a result of a cash-in operation. Note that for a reject cash unit (*type* is *rejectCassette*), this value is unreliable, since the typical reason for dumping items to the reject cash unit is a suspected count failure. For a retract cash unit (*type* is *retractCassette*) this value specifies the number of retract operations which result in items entering the cash unit.

Name	Type	Default	Description
maximum	integer		This field is only applicable to retract and reject cash units. When ulCount reaches this value the threshold event CashManagement.CashUnitThresholdEvent (*high*) will be generated. If this value is non-zero then hardware sensors in the device do not trigger threshold events. If this value is zero then hardware sensors will trigger threshold events if *hardwareSensor* is TRUE. This value is persistent.
status	string		Supplies the status of the cash unit. Following values are possible: "ok": The cash unit is in a good state. "full": The cash unit is full. "high": The cash unit is almost full (i.e. reached or exceeded the threshold defined by *maximum*). "low": The cash unit is almost empty (i.e. reached or below the threshold defined by *minimum*). "empty": The cash unit is empty, or insufficient items in the cash unit are preventing further dispense operations. "inoperative": The cash unit is inoperative. "missing": The cash unit is missing. "noValue": The values of the specified cash unit are not available. "noReference": There is no reference value available for the notes in this cash unit. The cash unit has not been calibrated. "manuelInsertion": The cash unit has been inserted (including removal followed by a reinsertion) when the device was not in the exchange state. This cash unit cannot be dispensed from.
appLock	boolean		If this value is TRUE items cannot be dispensed from or deposited into the cash unit. If this value is TRUE and the application attempts to use the cash unit a CashManagement.CashUnitErrorEvent event will be generated and an error completion message will be returned. This value is persistent.

Name	Type	Default	Description
physical	array		Array of pyhiscal cash unit objects.
cashUnitName	string		A name which helps to identify the logical type of the cash unit. This is especially useful in the case of cash units of type *document* where different documents can have the same currency and value. For example, travelers checks and bank checks may have the same currency and value but still need to be identifiable as different types of document. Where this value is not relevant (e.g. in bill cash units) the property can be omitted. This value is persistent.
initialCount	integer		Initial number of items contained in the cash unit. This value is persistent.
dispensedCount	integer		The number of items dispensed from all the physical cash units associated with this cash unit. This count is incremented when the items are removed from any of the associated physical cash units. This count includes any items that were rejected during the dispense operation and are no longer in this cash unit. This field is always zero for cash units with a *type* of *rejectCassette* or *retractCassette*. This value is persistent.
presentedCount	integer		The number of items dispensed from all the physical cash units associated with this cash unit. This count is incremented when the items are removed from any of the associated physical cash units. This count includes any items that were rejected during the dispense operation and are no longer in this cash unit. This field is always zero for cash units with a *type* of *rejectCassette* or *retractCassette*. This value is persistent.
retractedCount	integer		The number of items that have been accessible to a customer and retracted into all the physical cash units associated with this cash unit. This value is persistent.

Name	Type	Default	Description
rejectCount	integer		The number of items dispensed from this cash unit which have been rejected, are in a cash unit other than this cash unit, and which have not been accessible to a customer. This value may be unreliable, since a typical reason for rejecting items is a suspected pick failure. Other reasons for rejecting items may include incorrect note denominations, classifications not valid for dispensing, or where the transaction has been cancelled and a Reject command has been called. For reject and retract cash units (*type* is *rejectCassette* or *retractCassette*) this field does not apply and will be reported as zero. This value is persistent.
minimum	integer		This field is not applicable to retract and reject cash units. For all other cash units, when *count* reaches this value the threshold event CashManagement.CashUnitThresholdEvent (*low*) will be generated. If this value is non-zero then hardware sensors in the device do not trigger threshold events. If this value is zero then hardware sensors will trigger threshold events if *hardwareSensor* is TRUE. This value is persistent.
	object		
itemType	object		Specifies the type of items the cash unit takes as a combination of the following flags. The table in the Comments section of this command defines how to interpret the combination of these flags (TODO: include Table)
itemType.all	boolean		The cash unit takes all fit banknote types. These are level 4 notes which are fit for recycling.
itemType.unfit	boolean		The cash unit takes all unfit banknotes. These are level 4 notes which are unfit for recycling.
itemType.individual	boolean		The cash unit takes all types of fit banknotes specified in an individual list. These are level 4 notes which are fit for recycling.
itemType.level1	boolean		Level 1 note types are stored in this cash unit.

Name	Type	Default	Description
itemType.level2	boolean		If notes can be classified as level 2, then level 2 note types are stored in this cash unit.
itemType.level3	boolean		If notes can be classified as level 3, then level 3 note types are stored in this cash unit.
itemType.itemProcessor	boolean		The cash unit can accept items on the ItemProcessor interface.
itemType.unfitIndividual	boolean		The cash unit takes all types of unfit banknotes specified in an individual list. These are level 4 notes which are unfit for recycling.
cashInCount	integer		Count of items that have entered the logical cash unit. This counter is incremented whenever an item enters a physical cash unit that belongs to this logical cash unit for any reason, unless it originated from this cash unit but was returned without being accessible to a customer. For a retract cash unit this value represents the total number of items of all types in the cash unit, or if the device cannot count items during a retract operation this value will be zero. This value is persistent.
noteNumberList	object		Array of cash items inside the cash unit. The content of this structure is persistent. If the cash unit is Dispenser specific cash unit with *type billCassette* or the contents of the cash unit are not known this structure will be omitted. If the cash unit is of *type retractCassette* this pointer will be omitted except for the following cases: • If the retract cash unit is configured to accept level 2 notes then the number and type of level 2 notes is returned in the *noteNumberList* and *count* contains the number of retract operations. *cashInCount* contains the actual number of level 2 notes. • If items are recognized during retract operations then the number and type of notes retracted is returned in *noteNumberList* and *count* contains the number of retract operations. *cashInCount* contains the actual number of retracted items.

Name	Type	Default	Description
noteNumberList.noteNumber	array		Array of banknote numbers the cash unit contains.
noteNumberList.noteNumber.noteID	integer		Identification of note type. The Note ID represents the note identifiers reported by the *CashAcceptor.BanknoteTypes* command. If this value is zero then the note type is unknown.
noteNumberList.noteNumber.count	integer		Actual count of cash items. The value is incremented each time cash items are moved to a cash unit. In the case of recycle cash units this count is decremented as defined in the description of the logical *count* field.
noteIDs	array		Array of integers which contains the note IDs of the banknotes the cash-in cash unit or recycle cash unit can take. This field only applies to *individual* cassette types. If there are no note IDs defined for the cassette or the cassette is not defined as *individual* then *noteIDs* will be omitted.
extra	array		Pointer to a list of vendor-specific information about the logical cash unit. The information is returned as a series of “key=value” strings so that it is easily extensible by Service Providers.

Example Message (generated)

```
{
  "headers": {
    "requestId": "b34800d0-9dd2-4d50-89ea-92d1b13df54b",
    "type": "command",
    "name": "string"
  },
  "payload": {
    "number": 0,
    "type": "notApplicable",
    "unitID": "string",
    "currencyID": "string",
    "values": 0,
    "count": 0,
    "maximum": 0,
    "status": "ok",
    "appLock": true,
    "physical": [
      {
        "physicalPositionName": "string",
        "unitID": "string",
        "count": 0,
        "maximum": 0,
        "pStatus": "ok",
        "hardwareSensor": true,
        "initialCount": 0,
        "dispensedCount": 0,
        "presentedCount": 0,
        "retractedCount": 0,
        "rejectCount": 0,
        "cashInCount": 0,
        "extra": [
          "string"
        ]
      }
    ],
    "cashUnitName": "string",
    "initialCount": 0,
    "dispensedCount": 0,
    "presentedCount": 0,
    "retractedCount": 0,
    "rejectCount": 0,
    "minimum": 0,
    "itemType": {
      "all": true,
      "unfit": true,
      "individual": true,
      "level1": true,
      "level2": true,
      "level3": true,
      "itemProcessor": true,
      "unfitIndividual": true
    },
    "cashInCount": 0,
    "noteNumberList": {
      "noteNumber": [
        {
          "noteID": 0,
          "count": 0
        }
      ]
    },
    "noteIDs": [
      0
    ],
    "extra": [
      "string"
    ]
  }
}
```