# XFS4IoT Preview Messaging for CashManagement Draft 0.0.3 documentation

## Introduction

## Documentation

## Commands

## Unsolicited Events

## Events

- CashManagement.NoteErrorEvent
- CashManagement.InputP6Event
- CashManagement.InfoAvailableEvent
- CashAcceptor.ShutterStatusChangedEvent
- Dispenser.ItemsTakenEvent

# XFS4IoT Preview Messaging for CashManagement Draft 0.0.3

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

This interface is to be used together with Dispenser and/or CashAcceptor interfaces to handle management of cash units, cash counts and banknote information.

# Commands

## CashManagement.CashUnitInfo

### Description

This command is used to obtain information regarding the status and contents of cash units.

Where a logical cash unit is configured but there is no corresponding physical cash unit currently present in the device, information about the missing cash unit will still be returned in the *list* field of the completion message. The status of the cash unit will be reported as *missing*.

It is possible that one logical cash unit may be associated with more than one physical cash unit. In this case, the number of cash unit structures returned in *cashUnitInfo* will reflect the number of logical cash units. That is, if a system contains four physical cash units but two of these are treated as one logical cash unit, *cashUnitInfo* will contain information about the three logical cash units. Information about the physical cash unit(s) associated with a logical cash unit is contained in the *physical* objects representing the logical cash unit.

It is also possible that multiple logical cash units may be associated with one physical cash unit. This should only occur if the physical cash unit is capable of handling this situation, i.e. if it can store multiple denominations and report meaningful count and replenishment information for each denomination or if it can store retracted and rejected items as separate logical units and report meaningful count and replenishment information for each of them. In this case the information returned in *cashUnitInfo* will again reflect the number of logical cash units in the CDM.

### 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 |
| | 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",
        "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.TellerInfo

### Description

This command only applies to Teller CDMs. It allows the application to obtain counts for each currency assigned to the teller. These counts represent the total amount of currency dispensed by the teller in all transactions. This command also enables the application to obtain the position assigned to each teller. If the input parameter is NULL, this command will return information for all tellers and all currencies. The teller information is persistent.

### 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 the teller. If the value of *tellerID* is not valid the error *invalidTellerID* is reported. |
| currencyID | string | | Three character ISO format currency identifier [Ref 2]. |

**Example Message (generated)**

```
{
  "headers": {
    "requestId": "b34800d0-9dd2-4d50-89ea-92d1b13df54b",
    "type": "command",
    "name": "string"
  },
  "payload": {
    "timeout": "5000",
    "tellerID": 0,
    "currencyID": "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 |
| tellerDetails | array | | Array of teller detail objects. |
| tellerDetails.tellerID | integer | | Identification of the teller. |

| Name | Type | Default | Description |
|---|---|---|---|
| tellerDetails.inputPosition | string | | The input position assigned to the teller for cash entry. Following values are possible:<br>"none": No position is assigned to the teller.<br>"left": Left position is assigned to the teller.<br>"right": Right position is assigned to the teller.<br>"center": Center position is assigned to the teller.<br>"top": Top position is assigned to the teller.<br>"bottom": Bottom position is assigned to the teller.<br>"front": Front position is assigned to the teller.<br>"rear": Rear position is assigned to the teller. |
| tellerDetails.outputPosition | string | | The output position from which cash is presented to the teller. Following values are possible:<br>"none": No position is assigned to the teller.<br>"left": Left position is assigned to the teller.<br>"right": Right position is assigned to the teller.<br>"center": Center position is assigned to the teller.<br>"top": Top position is assigned to the teller.<br>"bottom": Bottom position is assigned to the teller.<br>"front": Front position is assigned to the teller.<br>"rear": Rear position is assigned to the teller. |
| tellerDetails.tellerTotals | array | | Array of teller total objects |

| Name | Type | Default | Description |
|---|---|---|---|
| tellerDetails.tellerTotals.currencyID | string | | Three character ISO format currency identifier [Ref. 2]. |
| tellerDetails.tellerTotals.itemsReceived | integer | | The total amount of items (other than coins) of the specified currency accepted. The amount is expressed in minimum dispense units (see section CashManagement.CurrencyExp). |
| tellerDetails.tellerTotals.itemsDispensed | integer | | The total amount of items (other than coins) of the specified currency dispensed. The amount is expressed in minimum dispense units (see section CashManagement.CurrencyExp). |
| tellerDetails.tellerTotals.coinsReceived | integer | | The total amount of coin currency accepted. The amount is expressed in minimum dispense units (see section CashManagement.CurrencyExp). |
| tellerDetails.tellerTotals.coinsDispensed | integer | | The total amount of coin currency dispensed. The amount is expressed in minimum dispense units (see section CashManagement.CurrencyExp). |
| tellerDetails.tellerTotals.cashBoxReceived | integer | | The total amount of cash box currency accepted. The amount is expressed in minimum dispense units (see section CashManagement.CurrencyExp). |
| tellerDetails.tellerTotals.cashBoxDispensed | integer | | The total amount of cash box currency dispensed. The amount is expressed in minimum dispense units (see section CashManagement.CurrencyExp). |

**Example Message (generated)**

```
{
  "headers": {
    "requestId": "b34800d0-9dd2-4d50-89ea-92d1b13df54b",
    "type": "command",
    "name": "string"
  },
  "payload": {
    "status": "ok",
    "errorDescription": "string",
    "tellerDetails": [
      {
        "tellerID": 0,
        "inputPosition": "none",
        "outputPosition": "none",
        "tellerTotals": [
          {
            "currencyID": "string",
            "itemsReceived": 0,
            "itemsDispensed": 0,
            "coinsReceived": 0,
            "coinsDispensed": 0,
            "cashBoxReceived": 0,
            "cashBoxDispensed": 0
          }
        ]
      }
    ]
  }
}
```

### Event Messages

## CashManagement.CurrencyExp

### Description

This command returns each exponent assigned to each currency known to the Service Provider.

### 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 |
| currencyExp | array | | Array of currency exponent objects. |
| currencyExp.currencyID | string | | Currency identifier in ISO 4217 format [Ref 2]. |
| currencyExp.exponent | integer | | Currency exponent in ISO 4217 format [Ref. 2]. |

#### Example Message (generated)

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

### Event Messages

## CashManagement.GetItemInfo

### Description

This command is used to get information about a single detected item**. This information is available from the point where the first** CashManagement.InfoAvailableEvent **event is generated until one of the following commands is executed:**

CashAcceptor.CashInStart, CashAcceptor.CashIn, CashAcceptor.CashInRollback, CashAcceptor.CashInEnd, CashAcceptor.Retract, CashAcceptor.Reset, CashAcceptor.CreateP6Signature, CashAcceptor.Replenish, CashAcceptor.CashUnitCount. Dispenser.Dispense, Dispenser.Count, Dispenser.Present, Dispenser.Retract, Dispenser.Reject, Dispenser.OpenShutter, Dispenser.CloseShutter, Dispenser.Reset, CashManagement.StartExchange, CashManagement.EndExchange, CashManagement.CalibrateCashUnit, Dispenser.TestCashUnits. This command is similar to the CashAcceptor.GetP6Signature command but returns additional information for level 2 / level 3 notes and also returns information relating to level 4 notes. The CashAcceptor.GetP6Info command, the CashAcceptor.GetP6Signature command and the CashManagement.InputP6Event event only relate to level 2 and level 3 notes. The CashManagement.InputP6Event event signals that a suspected forgery has been detected and is only generated when level 2 and/or level 3 notes are detected.

This command is used to retrieve the required information on an individual item basis. Applications should loop retrieving the information for each index and for each level reported with the CashManagement.InfoAvailableEvent **event**.

### 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 | Type | Default | Description |
|---|---|---|---|
| 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 | | |
| level | string | | Defines the note level. Following values are possible:<br>"level1": Information for level 1 notes. Only an image file can be retrieved for level 1 notes.<br>"level2": Information for level 2 notes. On systems that do not classify notes as level 2 this value cannot be used and "invalidData" will be returned.<br>"level3": Information for level 3 notes. On systems that do not classify notes as level 3 this value cannot be used and "invalidData" will be returned.<br>"level4": Information for level 4 notes. |
| index | integer | | Specifies the index for the item information required. |
| itemInfoType | object | | Specifies the type of information required. |
| itemInfoType.serialNumber | boolean | | Serial number of the item. |
| itemInfoType.signature | boolean | | Signature of the item. |
| itemInfoType.imageFile | boolean | | Image file of the item. |

**Example Message (generated)**

```
{
  "headers": {
    "requestId": "b34800d0-9dd2-4d50-89ea-92d1b13df54b",
    "type": "command",
    "name": "string"
  },
  "payload": {
    "timeout": "5000",
    "level": "level1",
    "index": 0,
    "itemInfoType": {
      "serialNumber": true,
      "signature": true,
      "imageFile": true
    }
  }
}
```

## 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 | | Currency ID in ISO 4217 format [Ref. 2]. This value will be omitted for level 1 items. |
| value | integer | | The value of a single item expressed in minimum dispense units. This value will be zero for level 1 items. |
| release | integer | | The release of the banknote type. The higher this number is, the newer the release. Zero means that there is only one release of that banknote type. This value has not been standardized and therefore a release number of the same banknote will not necessarily have the same value in different systems. This value will be zero for level 1 items. |
| | object | | |

| Name | Type | Default | Description |
|---|---|---|---|
| noteID | integer | | Identification of note type. This value will be zero for level 1 items. |
| | object | | |
| serialNumber | string | | This field contains the serial number of the item as a string. A '?' character (0x003F) is used to represent any serial number character that cannot be recognized. If no serial number is available or has not been requested then *serialNumber* is NULL. |
| p6Signature | object | | This field contains the signature for the item. If no signature is available or has not been requested then this field is omitted. |
| p6Signature.noteId | integer | | Identification of note type. |
| p6Signature.orientation | object | | Orientation of the entered banknote. |
| p6Signature.orientation.frontTop | boolean | | If note is inserted wide side as the leading edge, the note was inserted with the front image facing up and the top edge of the note was inserted first. If the note is inserted short side as the leading edge, the note was inserted with the front image face up and the left edge was inserted first. |
| p6Signature.orientation.frontBottom | boolean | | If note is inserted wide side as the leading edge, the note was inserted with the front image facing up and the bottom edge of the note was inserted first. If the note is inserted short side as the leading edge, the note was inserted with the front image face up and the right edge was inserted first. |
| p6Signature.orientation.backTop | boolean | | If note is inserted wide side as the leading edge, the note was inserted with the back image facing up and the top edge of the note was inserted first. If the note is inserted short side as the leading edge, the note was inserted with the back image face up and the left edge was inserted first. |
| p6Signature.orientation.backBottom | boolean | | If note is inserted wide side as the leading edge, the note was inserted with the back image facing up and the bottom edge of the note was inserted first. If the note is inserted short side as the leading edge, the note was inserted with the back image face up and the right edge was inserted first. |
| p6Signature.orientation.unknown | boolean | | The orientation for the inserted note can not be determined. |
| p6Signature.orientation.notSupported | boolean | | The hardware is not capable to determine the orientation. |

| Name | Type | Default | Description |
|---|---|---|---|
| p6Signature.signature | string | | Base64 encoded signature data. |
| imageFile | string | | Base64 encoded binary image data. If the Service does not support this function or the image file has not been requested then imageFile is omitted. |

**Example Message (generated)**

```
{
  "headers": {
    "requestId": "b34800d0-9dd2-4d50-89ea-92d1b13df54b",
    "type": "command",
    "name": "string"
  },
  "payload": {
    "status": "ok",
    "errorDescription": "string",
    "currencyID": "string",
    "value": 0,
    "release": 0,
    "serialNumber": "string",
    "p6Signature": {
      "noteId": 0,
      "orientation": {
        "frontTop": true,
        "frontBottom": true,
        "backTop": true,
        "backBottom": true,
        "unknown": true,
        "notSupported": true
      },
      "signature": "string"
    },
    "imageFile": "string"
  }
}
```

### Event Messages

## CashManagement.GetBlacklist

### Description

This command is used to retrieve the entire blacklist information preset inside the device or set via the CashManagement.SetBlacklist or CashManagement.SetClassificationList command.

### 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 |
| | object | | |

| Name | Type | Default | Description |
|---|---|---|---|
| version | string | | This is an application defined Unicode string that sets the version identifier of the blacklist. This property can be set to omitted if it has no version identifier. |
| blacklistElements | array | | Array of blacklist objects. |
| blacklistElements.serialNumber | string | | This string defines the serial number or a mask of serial numbers of one blacklist item with the defined currency and value. For a definition of the mask see Section 4 (TODO). |
| blacklistElements.currencyID | string | | The three character ISO format currency identifier [Ref. 2] of the blacklist element. |
| blacklistElements.value | integer | | The value of a blacklist element. This field can be zero to represent all values. |

#### Example Message (generated)

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

### Event Messages

## CashManagement.GetClassificationList

### Description

This command is used to retrieve the entire note classification information pre-set inside the device or set via the CashManagement.SetClassificationList or CashManagement.SetBlackist command. This extends the functionality provided by the blacklist commands and allows additional flexibility, for example to specify that notes can be taken out of circulation by specifying them as unfit. Any items not returned in this list will be handled according to normal classification rules.

## 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 |

| Name | Type | Default | Description |
|---|---|---|---|
| errorDescription | string | | If error, identified that cause of the error |
| | object | | |
| version | string | | This is an application defined string that sets the version identifier of the classification list. This property can be omitted if it has no version identifier. |
| classificationElements | array | | Array of classification objects. |
| classificationElements.serialNumber | string | | This string defines the serial number or a mask of serial numbers of one blacklist item with the defined currency and value. For a definition of the mask see Section 4 (TODO). |
| classificationElements.currencyID | string | | The three character ISO format currency identifier [Ref. 2] of the blacklist element. |
| classificationElements.value | integer | | The value of a blacklist element. This field can be zero to represent all values. |
| classificationElements.level | string | | Specifies the note level. Following values are possible:<br>"level1": The element specifies notes to be treated as level 1 notes.<br>"level2": The element specifies notes to be treated as level 2 notes.<br>"level3": The element specifies notes to be treated as level 3 notes.<br>"level4": The element specifies notes to be treated as level 4 notes. |

| Name | Type | Default | Description |
|---|---|---|---|
| classificationElements.unfit | boolean | | Specifies whether the item is to be treated as unfit for dispensing. Applies only where *level* is "level4". |

**Example Message (generated)**

```
{
  "headers": {
    "requestId": "b34800d0-9dd2-4d50-89ea-92d1b13df54b",
    "type": "command",
    "name": "string"
  },
  "payload": {
    "status": "ok",
    "errorDescription": "string",
    "version": "string",
    "classificationElements": [
      {
        "serialNumber": "string",
        "currencyID": "string",
        "value": 0,
        "level": "level1",
        "unfit": true
      }
    ]
  }
}
```

### Event Messages

## CashManagement.GetAllItemsInfo

### Description

This command can be used to retrieve all item information available for all levels at once by specifying "levelAll" in the *level* parameter. Or this command can be used to retrieve all information for a particular level of banknote. **This information is available from the point where the first** CashManagement.InfoAvailableEvent **event is generated until one of the following commands is executed:**

CashAcceptor.CashInStart, CashAcceptor.CashIn, CashAcceptor.CashInRollback, CashAcceptor.CashInEnd, CashAcceptor.Retract, CashAcceptor.Reset, CashAcceptor.CreateP6Signature, CashAcceptor.Replenish, CashAcceptor.CashUnitCount. Dispenser.Dispense, Dispenser.Count, Dispenser.Present, Dispenser.Retract, Dispenser.Reject, Dispenser.OpenShutter, Dispenser.CloseShutter, Dispenser.Reset, CashManagement.StartExchange, CashManagement.EndExchange, CashManagement.CalibrateCashUnit, Dispenser.TestCashUnits. This command is similar to the CashAcceptor.GetP6Signature command but returns additional information for level 2 / level 3 notes and also returns information relating to level 4 notes. The CashAcceptor.GetP6Info command, the CashAcceptor.GetP6Signature command and the CashManagement.InputP6Event event only relate to level 2 and level 3 notes. The CashManagement.InputP6Event event signals that a suspected forgery has been detected and is only generated when level 2 and/or level 3 notes are detected.

## 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 |
|---|---|---|---|
| level | string | | Defines the note level. Following values are possible:<br>"level1": Information for level 1 notes. Only an image file can be retrieved for level 1 notes.<br>"level2": Information for level 2 notes is to be returned with the allItemsInfo output parameter. On systems that do not classify notes as level 2 this value cannot be used and "invalidData" will be returned.<br>"level3": Information for level 3 notes is to be returned with the allItemsInfo output parameter. On systems that do not classify notes as level 3 this value cannot be used and "invalidData" will be returned.<br>"level4": Information for level 4 notes is to be returned with the allItemsInfo output parameter.<br>"levelAll": Information for all levels all items is to be returned with the allItemsInfo output parameter. |

#### Example Message (generated)

```
{
  "headers": {
    "requestId": "b34800d0-9dd2-4d50-89ea-92d1b13df54b",
    "type": "command",
    "name": "string"
  },
  "payload": {
    "timeout": "5000",
    "level": "level1"
  }
}
```

## 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 | | |
| itemsList | array | | Array of "all item info" objects. |

#### Example Message (generated)

```
{
  "headers": {
    "requestId": "b34800d0-9dd2-4d50-89ea-92d1b13df54b",
    "type": "command",
    "name": "string"
  },
  "payload": {
    "status": "ok",
    "errorDescription": "string",
    "itemsList": [
      {
        "currencyID": "string",
        "value": 0,
        "release": 0,
        "level": "level1",
        "serialNumber": "string",
        "orientation": {
          "frontTop": true,
          "frontBottom": true,
          "backTop": true,
          "backBottom": true,
          "unknown": true,
          "notSupported": true
        },
        "p6SignatureFile": "string",
        "imageFile": "string",
        "onBlacklist": "onBlacklist",
        "itemLocation": "device",
        "number": 0,
        "onClassificationList": "onClassificationList",
        "itemDeviceLocation": "stacker"
      }
    ]
  }
}
```

### Event Messages

## CashManagement.SetTellerInfo

### Description

This command allows the application to initialize counts for each currency assigned to the teller. The values set by this command are persistent. This command only applies to Teller ATMs.

### 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. |
| | object | | |
| action | string | | The action to be performed. Following values are possible:<br>"createTeller": A teller is to be added.<br>"modifyTeller": Information about an existing teller is to be modified.<br>"deleteTeller": A teller is to be removed. |
| tellerDetails | object | | Teller details object. |
| tellerDetails.tellerID | integer | | Identification of the teller. |

| Name | Type | Default | Description |
|---|---|---|---|
| tellerDetails.inputPosition | string | | The input position assigned to the teller for cash entry. Following values are possible:<br>"none": No position is assigned to the teller.<br>"left": Left position is assigned to the teller.<br>"right": Right position is assigned to the teller.<br>"center": Center position is assigned to the teller.<br>"top": Top position is assigned to the teller.<br>"bottom": Bottom position is assigned to the teller.<br>"front": Front position is assigned to the teller.<br>"rear": Rear position is assigned to the teller. |
| tellerDetails.outputPosition | string | | The output position from which cash is presented to the teller. Following values are possible:<br>"none": No position is assigned to the teller.<br>"left": Left position is assigned to the teller.<br>"right": Right position is assigned to the teller.<br>"center": Center position is assigned to the teller.<br>"top": Top position is assigned to the teller.<br>"bottom": Bottom position is assigned to the teller.<br>"front": Front position is assigned to the teller.<br>"rear": Rear position is assigned to the teller. |
| tellerDetails.tellerTotals | array | | Array of teller total objects |

| Name | Type | Default | Description |
|---|---|---|---|
| tellerDetails.tellerTotals.currencyID | string | | Three character ISO format currency identifier [Ref. 2]. |
| tellerDetails.tellerTotals.itemsReceived | integer | | The total amount of items (other than coins) of the specified currency accepted. The amount is expressed in minimum dispense units (see section CashManagement.CurrencyExp). |
| tellerDetails.tellerTotals.itemsDispensed | integer | | The total amount of items (other than coins) of the specified currency dispensed. The amount is expressed in minimum dispense units (see section CashManagement.CurrencyExp). |
| tellerDetails.tellerTotals.coinsReceived | integer | | The total amount of coin currency accepted. The amount is expressed in minimum dispense units (see section CashManagement.CurrencyExp). |
| tellerDetails.tellerTotals.coinsDispensed | integer | | The total amount of coin currency dispensed. The amount is expressed in minimum dispense units (see section CashManagement.CurrencyExp). |
| tellerDetails.tellerTotals.cashBoxReceived | integer | | The total amount of cash box currency accepted. The amount is expressed in minimum dispense units (see section CashManagement.CurrencyExp). |
| tellerDetails.tellerTotals.cashBoxDispensed | integer | | The total amount of cash box currency dispensed. The amount is expressed in minimum dispense units (see section CashManagement.CurrencyExp). |

**Example Message (generated)**

```
{
  "headers": {
    "requestId": "b34800d0-9dd2-4d50-89ea-92d1b13df54b",
    "type": "command",
    "name": "string"
  },
  "payload": {
    "timeout": "5000",
    "action": "createTeller",
    "tellerDetails": {
      "tellerID": 0,
      "inputPosition": "none",
      "outputPosition": "none",
      "tellerTotals": [
        {
          "currencyID": "string",
          "itemsReceived": 0,
          "itemsDispensed": 0,
          "coinsReceived": 0,
          "coinsDispensed": 0,
          "cashBoxReceived": 0,
          "cashBoxDispensed": 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

- CashManagement.TellerInfoChangedEvent

## CashManagement.SetCashUnitInfo

### Description

This command is used to adjust information about the status and contents of the cash units present in the ATM.

This command generates the CashManagement.CashUnitInfoCahngedEvent to inform applications that cash unit information has been changed.

This command can only be used to change software counters, thresholds and the application lock. All other fields in the input structure will be ignored.

The following fields of the may be updated by this command:

*count*
*cashInCount*
*maximum*
*appLock*
*noteNumberList (contents must be consistent with ulCount)*
*initialCount*
*dispensedCount*
*presentedCount*
*retractedCount*
*rejectCount*
*minimum*

As may the following fields of the physical structure:

*cashInCount*
*count*
*initialCount*
*dispensedCount*
*presentedCount*
*retractedCount*
*rejectCount*

Any other changes must be performed via an exchange operation.

The *physical* counts must be consistent with the logical cash unit counts. The Service controls whether the logical counts are maintained separately or are based on the sum of the physical counts.

If the fields *count* and *cashInCount* of *physical* are set to zero by this command, the application is indicating that it does not wish counts to be maintained for the physical cash units. Counts on the logical cash units will still be maintained and can be used by the application. If the physical counts are set by this command then the logical count will be the sum of the physical counts and any value sent as a logical count will be ignored.

The values set by this command are persistent.

## 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 | | |
| list | array | | Array of cash unit objects. |

#### Example Message (generated)

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

### 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

- CashManagement.CashUnitThresholdEvent
- CashManagement.CashUnitInfoChangedEvent
- CashManagement.CashUnitErrorEvent

## CashManagement.StartExchange

### Description

This command puts the ATM in an exchange state, i.e. a state in which cash units can be emptied, replenished, removed or replaced. Other than the updates which can be made via the CashManagement.SetCashUnitInfo command all changes to a cash unit must take place while the cash unit is in an exchange state.

The command returns current cash unit information in the form described in the documentation of the CashManagement.CsahUnitInfo command. This command will also initiate any physical processes which may be necessary to make the cash units accessible. Before using this command an application should first have obtained exclusive control of the CashManagement interface.

This command may return a successful completion even if CashManagement.CashUnitErrorEvents are generated. If this command returns a successful completion the ATM is in an exchange state.

While in an exchange state the ATM will process all requests, excluding cash related commands other than CashManagement.EndExchange, Dispenser.SetMixTable and Reset commands (e.g. Dispenser.Reset).

Any other command will result in the error "exchangeActive" being generated.

If an error is returned by this command, the CashManagement.CashUnitInfo command should be used to determine the cash unit 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 | | |

| Name | Type | Default | Description |
|---|---|---|---|
| exchangeType | string | | Specifies the type of cash unit exchange operation. Following values are possible:<br>"byHand": The cash units will be replenished manually either by filling or emptying the cash unit by hand or by replacing the cash unit.<br>"toCassettes": Items will be moved from the replenishment container to the bill cash units.<br>"clearRecycler": Items will be moved from a recycle cash unit to a cash unit or output position.<br>"depositInto": Items will be moved from the deposit entrance to the bill cash units. See section 8.16 (TODO) for an example flow. |
| tellerID | integer | | Identifies the teller. If the device is a Self-Service ATM this field is ignored. |
| cUNumList | array | | Array of integers containing the logical numbers of the cash units to be exchanged. |
| output | object | | This field is used when the *exchangeType* is "clearRecycler", i.e. a recycle cash unit is to be emptied. |
| output.logicalNumber | integer | | Logical number of recycle cash unit be emptied. |
| output.position | object | | Determines to which position the cash should be moved as a combination of the following flags: |
| output.position.default | boolean | | Move items to a cash unit. If no cash unit is specified in *number*, use the default output position. |

| Name | Type | Default | Description |
|---|---|---|---|
| output.position.left | boolean | | Move items to the left output position. |
| output.position.right | boolean | | Move items to the right output position. |
| output.position.center | boolean | | Move items to the center output position. |
| output.position.top | boolean | | Move items to the top output position. |
| output.position.bottom | boolean | | Move items to the bottom output position. |
| output.position.front | boolean | | Move items to the front output position. |
| output.position.rear | boolean | | Move items to the rear output position. |
| output.number | integer | | Logical number of the cash unit the items are to be moved to. |

**Example Message (generated)**

```
{
  "headers": {
    "requestId": "b34800d0-9dd2-4d50-89ea-92d1b13df54b",
    "type": "command",
    "name": "string"
  },
  "payload": {
    "timeout": "5000",
    "exchangeType": "byHand",
    "tellerID": 0,
    "cUNumList": [
      0
    ],
    "output": {
      "logicalNumber": 0,
      "position": {
        "default": true,
        "left": true,
        "right": true,
        "center": true,
        "top": true,
        "bottom": true,
        "front": true,
        "rear": true
      },
      "number": 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 | | |
| 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",
        "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.CashUnitErrorEvent
- CashManagement.NoteErrorEvent
- CashManagement.CashUnitThresholdEvent
- CashManagement.CashUnitInfoChangedEvent
- CashManagement.InputP6Event
- CashManagement.InfoAvailableEvent
- CashAcceptor.ShutterStatusChangedEvent

## CashManagement.EndExchange

### Description

This command will end the exchange state. If any physical action took place as a result of the CashManagement.StartExchange command then this command will cause the cash units to be returned to their normal physical state, including depositing any remaining items where *exchangeType* is "depositInto". Any necessary device testing will also be initiated. The application can also use this command to update cash unit information in the form described in the documentation of the CashManagement.CashUnitInfo command.

The input parameters to this command may be ignored if the Service can obtain cash unit information from self-configuring cash units.

The *physical* counts must be consistent with the logical cash unit counts. The Service controls whether the logical counts are maintained separately or are based on the sum of the physical counts.

If the fields *count*, and *cashInCount* of *physical* are set to zero by this command, the application is indicating that it does not wish counts to be maintained for the physical cash units. Counts on the logical cash units will still be maintained and can be used by the application. If the physical counts are set by this command then the logical count will be the sum of the physical counts and any value sent as a logical count will be ignored.

If an error occurs during the execution of this command, then the application must issue a CashManagement.CashUnitInfo to determine the cash unit information.

**A CashManagement.CashUnitErrorEvent will be sent for any logical cash unit which cannot be successfully updated. If no cash units could be updated then a error code will be returned and CashManagement.CashUnitErrorEvent events generated for every logical cash unit that could not be updated.**

Even if this command does not return a successful completion the exchange state has ended.

## 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 | | |
| list | array | | Array of cash unit objects. |

#### Example Message (generated)

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

## 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<br>(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.CashUnitInfoChangedEvent
- CashManagement.CashUnitErrorEvent
- CashManagement.NoteErrorEvent
- CashManagement.InputP6Event
- CashManagement.InfoAvailableEvent

## CashManagement.CalibrateCashUnit

### Description

This command will cause a vendor dependent sequence of hardware events which will calibrate one or more physical cash units associated with a logical cash unit. This is necessary if a new type of bank note is put into the cash unit as the command enables the ATM to obtain the measures of the new bank notes.

If more than one physical cash unit is associated with the cash unit, it is up to the Service to determine whether all the physical cash units need to be calibrated or if it is sufficient to calibrate for one physical unit and load the data into the others.

This command cannot be used to calibrate cash units which have been locked by the application. A error code will be returned and a CashManagement.CashUnitErrorEvent generated.

### 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 | | |
| number | integer | | The logical number of the cash unit. |
| numOfBills | integer | | The number of bills to be dispensed during the calibration process. |
| position | object | | Specifies where the dispensed items should be moved to. |
| position.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. |
| position.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 |
|---|---|---|---|
| position.retractArea.outputPosition | string | | Output position from which to retract the items. Following values are possible:<br>"default": The default configuration information should be used.<br>"left": Retract items from the left output position.<br>"right": Retract items from the right output position.<br>"center": Retract items from the center output position.<br>"top": Retract items from the top output position.<br>"bottom": Retract items from the bottom output position.<br>"front": Retract items from the front output position.<br>"rear": Retract items from the rear output position. |
| position.retractArea.retractArea | string | | This value specifies the area to which the items are to be retracted. Following values are possible:<br>"retract": Retract the items to a retract cash unit.<br>"transport": Retract the items to the transport.<br>"stacker": Retract the items to the intermediate stacker area.<br>"reject": Retract the items to a reject cash unit.<br>"itemCassette": Retract the items to the item cassettes, i.e. cassettes that can be dispensed from. |

| Name | Type | Default | Description |
|---|---|---|---|
| position.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. |
| position.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:<br>"default": The default configuration.<br>"left": The left output position.<br>"right": The right output position.<br>"center": The center output position.<br>"top": The top output position.<br>"bottom": The bottom output position.<br>"front": The front output position.<br>"rear": The rear output position. |

**Example Message (generated)**

```
{
  "headers": {
    "requestId": "b34800d0-9dd2-4d50-89ea-92d1b13df54b",
    "type": "command",
    "name": "string"
  },
  "payload": {
    "timeout": "5000",
    "number": 0,
    "numOfBills": 0,
    "position": {
      "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 |
| | object | | |
| number | integer | | The logical number of the cash unit. |
| numOfBills | integer | | The number of bills to be dispensed during the calibration process. |

| Name | Type | Default | Description |
|---|---|---|---|
| position | object | | Specifies where the dispensed items should be moved to. |
| position.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. |
| position.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. |
| position.retractArea.outputPosition | string | | Output position from which to retract the items. Following values are possible:<br>"default": The default configuration information should be used.<br>"left": Retract items from the left output position.<br>"right": Retract items from the right output position.<br>"center": Retract items from the center output position.<br>"top": Retract items from the top output position.<br>"bottom": Retract items from the bottom output position.<br>"front": Retract items from the front output position.<br>"rear": Retract items from the rear output position. |

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

**Example Message (generated)**

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

## Event Messages

- CashManagement.CashUnitThresholdEvent
- CashManagement.CashUnitInfoChangedEvent
- CashManagement.CashUnitErrorEvent
- CashManagement.NoteErrorEvent
- CashManagement.InputP6Event
- CashManagement.InfoAvailableEvent

- Dispenser.ItemsTakenEvent

# CashManagement.SetBlacklist

## Description

This command is used to set all blacklist information. This list is persistent. Information set by this command overrides any existing blacklist or classification list, although it is not recommended that an application use both this command and CashManagement.SetClassificationList to avoid overlap and confusion.

## 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 | | |
| version | string | | This is an application defined Unicode string that sets the version identifier of the blacklist. This property can be set to omitted if it has no version identifier. |
| blacklistElements | array | | Array of blacklist objects. |
| blacklistElements.serialNumber | string | | This string defines the serial number or a mask of serial numbers of one blacklist item with the defined currency and value. For a definition of the mask see Section 4 (TODO). |
| blacklistElements.currencyID | string | | The three character ISO format currency identifier [Ref. 2] of the blacklist element. |
| blacklistElements.value | integer | | The value of a blacklist element. This field can be zero to represent all values. |

**Example Message (generated)**

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

## CashManagement.SetClassificationList

### Description

This command is used to specify the entire note classification list. Any items not specified in this list will be handled according to normal classification rules. This information is persistent. Information set by this command overrides any existing blacklist or classification list, although it is not recommended that an application use both this command and CashManagement.SetBlacklist to avoid overlap and confusion. If a note is reclassified, it is handled as though it was a note of the new classification. For example, a fit note reclassified as unfit would be treated as though it were unfit, which may mean that the note is not dispensed. Reclassification cannot be used to change a note's classification to a higher level, for example, a note recognized as counterfeit by the device cannot be reclassified as genuine. In addition, it is not possible to re-classify a level 2 note as level 1. If two or more classification elements specify overlapping note definitions, but different usLevel values then the first one takes priority.

### 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 | | |
| version | string | | This is an application defined string that sets the version identifier of the classification list. This property can be omitted if it has no version identifier. |
| classificationElements | array | | Array of classification objects. |

| Name | Type | Default | Description |
|---|---|---|---|
| classificationElements.serialNumber | string | | This string defines the serial number or a mask of serial numbers of one blacklist item with the defined currency and value. For a definition of the mask see Section 4 (TODO). |
| classificationElements.currencyID | string | | The three character ISO format currency identifier [Ref. 2] of the blacklist element. |
| classificationElements.value | integer | | The value of a blacklist element. This field can be zero to represent all values. |
| classificationElements.level | string | | Specifies the note level. Following values are possible:<br>"level1": The element specifies notes to be treated as level 1 notes.<br>"level2": The element specifies notes to be treated as level 2 notes.<br>"level3": The element specifies notes to be treated as level 3 notes.<br>"level4": The element specifies notes to be treated as level 4 notes. |
| classificationElements.unfit | boolean | | Specifies whether the item is to be treated as unfit for dispensing. Applies only where *level* is "level4". |

**Example Message (generated)**

```
{
  "headers": {
    "requestId": "b34800d0-9dd2-4d50-89ea-92d1b13df54b",
    "type": "command",
    "name": "string"
  },
  "payload": {
    "timeout": "5000",
    "version": "string",
    "classificationElements": [
      {
        "serialNumber": "string",
        "currencyID": "string",
        "value": 0,
        "level": "level1",
        "unfit": true
      }
    ]
  }
}
```

## Completion Message

### Message Header

| Name | Type | Default | Description |
|---|---|---|---|
| requestId<br>(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<br>(Required) | string | | The message type, either command, response, event or completion. |
| name<br>(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

## 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 |
|---|---|---|---|
| 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 |
|---|---|---|---|
| | 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:<br>"notApplicable": Not applicable. Typically means cash unit is missing.<br>"rejectCassette": Reject cash unit. This type will also indicate a combined reject/retract cash unit.<br>"billCassette": Cash unit containing bills.<br>"coinCylinder": Coin cylinder.<br>"coinDispenser": Coin dispenser as a whole unit.<br>"retractCassette": Retract cash unit.<br>"coupon": Cash unit containing coupons or advertising material.<br>"document": Cash unit containing documents.<br>"replenishmentContainer": Replenishment container. A cash unit can be refilled from a replenishment container.<br>"recycling": Recycling cash unit. This unit is only present when the device implements the Dispenser and CashAcceptor interfaces.<br>"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:<br>"ok": The cash unit is in a good state.<br>"full": The cash unit is full.<br>"high": The cash unit is almost full (i.e. reached or exceeded the threshold defined by *maximum*).<br>"low": The cash unit is almost empty (i.e. reached or below the threshold defined by *minimum*).<br>"empty": The cash unit is empty, or insufficient items in the cash unit are preventing further dispense operations.<br>"inoperative": The cash unit is inoperative.<br>"missing": The cash unit is missing.<br>"noValue": The values of the specified cash unit are not available.<br>"noReference": There is no reference value available for the notes in this cash unit. The cash unit has not been calibrated.<br>"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:<br>• 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.<br>• 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"
    ]
  }
}
```

## CashManagement.TellerInfoChangedEvent

### Description

This service event is generated when the counts assigned to a teller have changed. This event is only returned as a result of a CashManagement.SetTellerInfo 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 |
|---|---|---|---|
| tellerID | integer | | Integer holding the ID of the teller whose counts have changed. |

**Example Message (generated)**

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

# CashManagement.CountsChangedEvent

## Description

Deprecated

**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 |
|---|---|---|---|
| cUNumList | array | | Array of the number values of the cash units whose counts have changed. |

**Example Message (generated)**

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

## 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 | | |

| Name | Type | Default | Description |
|---|---|---|---|
| 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:<br>"notApplicable": Not applicable. Typically means cash unit is missing.<br>"rejectCassette": Reject cash unit. This type will also indicate a combined reject/retract cash unit.<br>"billCassette": Cash unit containing bills.<br>"coinCylinder": Coin cylinder.<br>"coinDispenser": Coin dispenser as a whole unit.<br>"retractCassette": Retract cash unit.<br>"coupon": Cash unit containing coupons or advertising material.<br>"document": Cash unit containing documents.<br>"replenishmentContainer": Replenishment container. A cash unit can be refilled from a replenishment container.<br>"recycling": Recycling cash unit. This unit is only present when the device implements the Dispenser and CashAcceptor interfaces.<br>"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. |

| Name | Type | Default | Description |
|---|---|---|---|
| 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. |
| 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:<br>"ok": The cash unit is in a good state.<br>"full": The cash unit is full.<br>"high": The cash unit is almost full (i.e. reached or exceeded the threshold defined by *maximum*).<br>"low": The cash unit is almost empty (i.e. reached or below the threshold defined by *minimum*).<br>"empty": The cash unit is empty, or insufficient items in the cash unit are preventing further dispense operations.<br>"inoperative": The cash unit is inoperative.<br>"missing": The cash unit is missing.<br>"noValue": The values of the specified cash unit are not available.<br>"noReference": There is no reference value available for the notes in this cash unit. The cash unit has not been calibrated.<br>"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:<br>• 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.<br>• 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"
    ]
  }
}
```

# 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:<br>"empty": Specified cash unit is empty.<br>"error": Specified cash unit has malfunctioned.<br>"full": Specified cash unit is full.<br>"locked": Specified cash unit is locked.<br>"invalid": Specified cash unit is invalid.<br>"config": An attempt has been made to change the settings of a self-configuring cash unit.<br>"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:<br>"notApplicable": Not applicable. Typically means cash unit is missing.<br>"rejectCassette": Reject cash unit. This type will also indicate a combined reject/retract cash unit.<br>"billCassette": Cash unit containing bills.<br>"coinCylinder": Coin cylinder.<br>"coinDispenser": Coin dispenser as a whole unit.<br>"retractCassette": Retract cash unit.<br>"coupon": Cash unit containing coupons or advertising material.<br>"document": Cash unit containing documents.<br>"replenishmentContainer": Replenishment container. A cash unit can be refilled from a replenishment container.<br>"recycling": Recycling cash unit. This unit is only present when the device implements the Dispenser and CashAcceptor interfaces.<br>"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:<br>"ok": The cash unit is in a good state.<br>"full": The cash unit is full.<br>"high": The cash unit is almost full (i.e. reached or exceeded the threshold defined by *maximum*).<br>"low": The cash unit is almost empty (i.e. reached or below the threshold defined by *minimum*).<br>"empty": The cash unit is empty, or insufficient items in the cash unit are preventing further dispense operations.<br>"inoperative": The cash unit is inoperative.<br>"missing": The cash unit is missing.<br>"noValue": The values of the specified cash unit are not available.<br>"noReference": There is no reference value available for the notes in this cash unit. The cash unit has not been calibrated.<br>"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:<br>• 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.<br>• 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.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 |
|---|---|---|---|
| reason | string | | The reason for the notes detection error. Following values are possible:<br>"doubleNote": Double notes have been detected.<br>"longNote": A long note has been detected.<br>"skewedNote": A skewed note has been detected.<br>"incorrectCount": An item counting error has occurred.<br>"notesTooClose": Notes have been detected as being too close.<br>"otherNoteError": An item error not covered by the other values has been detected.<br>"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 |
|---|---|---|---|
| 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:<br>"level2": Information for level 2 notes.<br>"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. |

| Name | Type | Default | Description |
|---|---|---|---|
| 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 |
|---|---|---|---|

| 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. |
| itemInfoSummary.level | string | | Defines the note level. Following values are possible:<br>"level1": Information for level 1 notes.<br>"level2": Information for level 2 notes.<br>"level3": Information for level 3 notes.<br>"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
      }
    ]
  }
}
```

# CashAcceptor.ShutterStatusChangedEvent

## Description

Within the limitations of the hardware sensors this service 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 |
|---|---|---|---|
| position | integer | | TODO |
| shutter | integer | | TODO |

#### Example Message (generated)

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

# 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 |
|---|---|---|---|
| position | string | | The output position from which the items have been removed. Following values are possible:<br>"default": The default configuration.<br>"left": The left output position.<br>"right": The right output position.<br>"center": The center output position.<br>"top": The top output position.<br>"bottom": The bottom output position.<br>"front": The front output position.<br>"rear": The rear output position. |

#### Example Message (generated)

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