

Reference Tables (v1 — Legacy)

fiskaltrust.Middleware — legacy POSSystem API v1 reference

Legacy documentation

This is legacy documentation. These reference tables describe the POSSystem API **v1** tagging structure. New integrations should use the current **v2** Reference Tables at <https://docs.fiskaltrust.eu/docs/poscreators/middleware-doc/general/reference-tables>. This document is provided for the reference of existing v1 integrations only.

Service Status: ftState

The ftState is returned with every receipt response. Through this status, fiskaltrust.Middleware can signal its operability or request processing logic.

The value 0xFFFF000000000000 indicates that the fiskaltrust.Middleware is available without reporting problems or notifications - it's a kind of a “ready” state. The XXXX contains the country specific code for which the service has been configured. The country specific code, is made of the country's code value following the ISO-3166-1-ALPHA-2 standard, converted from ASCII into hex.

The table below describes supported statuses for the ftState field. Those codes can be combined by using the logic operator OR.

Value	Description	Middleware Version
0xFFFF000000000001	“out of service” — The security mechanism has not yet been activated or was already deactivated.	1.0
0xFFFF000000000008	Late Signing Mode — End of Fail Receipt required	1.0

Table 1. Supported statuses signalled through the ftState field.

Example of reading ftState parameter

The following example shows how to extract the value of a flag into the ftState property.

```
if ((ReceiptResponse.ftState & 0x0000000000000001) != 0)
{
    //your code in case of out of service condition
}

if ((ReceiptResponse.ftState & 0x0000000000000008) != 0)
{
    //your code in case of Late Signing condition
}

if ((ReceiptResponse.ftState & 0x0000000000000040) != 0)
{
    //your code in case of Message Pending condition
}
```

Type of Receipt: `ftReceiptCase`

The `ftReceiptCase` indicates the receipt type and defines how it should be processed by the `fiskaltrust.SecurityMechanism`. The data type is `Int64` and contains the country specific code, which is a value following the ISO-3166-1-ALPHA-2 standard converted from ASCII into hex and used as byte 8 and 7. For definitions regarding national laws, please refer to the appropriate appendix.

Value	Description	Middleware Version
0x0000000000000000	“default value” — Unknown type of receipt. Automatic processing through the localization setting of the <code>fiskaltrust.Middleware</code> is attempted.	1.1

Table 2. Default value of the `ftReceiptCase` field.

`ftReceiptCaseFlag`

Business transactions can result in combinations of receipt types, which would be indicated using codes in bytes 6, 5, 4 and 3. These codes can be combined using the logic operator `OR`.

Value	Description	Middleware Version
0x0000000000001000	“failed receipt” — If a <code>ReceiptRequest</code> includes this flag, it sets the <code>fiskaltrust.Middleware</code> in a “late signing mode”. In order to leave this mode, an “end of failure” Receipt Request has to be made by the cash register terminal using a Zero Receipt. This can be necessary e.g. after a power or server outage. See the “Zero Receipt” section (https://docs.fiskaltrust.eu/docs/poscreators/middleware-doc/general/cash-register-integration#zero-receipt).	1.0
0x0000000000002000	“training receipt” — Used to separate the normal usage of the cash register system from the training mode. This flag can be added to each Receipt Request when training/testing is performed. This receipt will not produce any tax relevant changes.	1.0
0x0000000000004000	“reverse receipt” or “voided receipt” — Used to separate regular receipts from the receipts which were voided by setting the negative values for both Amount and Quantity in <code>Chargeitems</code> and <code>Payitems</code> of the receipt. The <code>cbPreviousReceiptReference</code> should be set to the <code>ReceiptReference</code> of the Receipt which should be voided.	1.0
0x0000000000008000	“handwritten receipt” — The transferred receipt contains data which has been collected in a handwritten receipt. There is no requirement for a precise time annotation on a handwritten receipt; we recommend using 12:00 for this purpose.	1.0
0x0000800000000000	“receipt request” — Used to retrieve an already processed receipt from the <code>fiskaltrust.Middleware</code> using the <code>cbReceiptReference</code> field. The <code>cbTerminalID</code> can also be included in this request. <code>Chargeitems</code> and <code>payitems</code> have to be exactly the same as in the requested receipt. If a matching receipt is found, its content will be returned. If nothing is found a null value is returned. This can be necessary if a communication problem occurs while <code>fiskaltrust.Middleware</code> processes a request.	1.1

Value	Description	Middleware Version
	To prevent a duplication of requested receipt, the cash register terminal can place an additional parameter inside the queue to influence the behavior of this “receipt request” when no receipt is found. Parameter name: “receiptrequestmode”. Parameter values: 0 (default) ... null is returned; 1 ... request is handled in the same way as new request. If processing is successful, the created ReceiptResponse is returned.	1.2

Table 3. Receipt-type flags that can be combined in the *ftReceiptCase* field.

These flags are based on national laws and regulations, for further information please refer to the appropriate country specific appendix.

Type of Service: *ftChargeItemCase*

The *ftChargeItemCase* defines the type of service or item in the charge item block and thus how the *fiskaltrust.SecurityMechanism* processes the individual receipts with regards to receipt generation. The data type is *Int64* and contains a country specific code which is a value following the ISO-3166-1-ALPHA-2 standard, converted from ASCII into hex and used as byte 8 and 7.

For definitions regarding national laws, please refer to the appropriate country specific appendix.

Value	Description	Middleware Version
0xXXXX000000000000	“default value” — Unknown type of service: Automatic processing through the localization setting of the <i>fiskaltrust.Middleware</i> is attempted.	1.0

Table 4. Default value of the *ftChargeItemCase* field.

Type of Payment: *ftPayItemCase*

The *ftPayItemCase* indicates the type of payment within the pay items block and defines how the *fiskaltrust.SecurityMechanism* processes the individual payment in terms of the receipt. The data type is *Int64* and contains a country specific code which is a value following the ISO-3166-1-ALPHA-2 standard, converted from ASCII into hex and used as byte 8 and 7.

For definitions regarding national laws, please refer to the appropriate appendix.

Value	Description	Middleware Version
0xXXXX000000000000	“default value” — unknown payment type: Automatic processing through the localization setting of the <i>fiskaltrust.Middleware</i> is attempted.	1.0

Table 5. Default value of the *ftPayItemCase* field.

Format of Signature: *ftSignatureFormat*

The *ftSignatureFormat* tells the cash register or input station which display format is required for the signature block section on the receipt.

Value	Description	Middleware Version
0x00	Unknown / no format defined	1.0
0x01	Text	1.0
0x02	Link	1.0
0x03	QR-Code (2D Code)	1.0
0x04	Code128 (Barcode)	1.0
0x05	OCR-A (optical character recognition, possible for Base32 data)	1.0
0x06	PDF417-(2D Code)	1.0
0x07	DATAMATRIX-(2D Code)	1.0
0x08	AZTEC-(2D Code)	1.0
0x09	EAN-8 (Barcode)	1.0
0x0A	EAN-13 (Barcode)	1.0
0x0B	UPC-A (Barcode)	1.0
0x0C	Code39 (Barcode, possible for Base32 data)	1.0
0x0D	Base64	1.3

Table 6. Display formats for the signature block defined by *ftSignatureFormat*.

Type of Signature: *ftSignatureType*

The *ftSignatureType* indicates type and origin of the signature. The data type is **Int64** and can contain a country specific code which is a value following the ISO-3166-1-ALPHA-2 standard, converted from ASCII into hex and used as byte 8 and 7.

For definitions regarding national laws, please refer to the appropriate appendix.

Value	Description	Middleware Version
0x0000000000000000	unknown	1.0
0x0000000000000100	information notification	1.0
0x0000000000000200	alert notification	1.0
0x0000000000000300	failure notification	1.0

Table 7. Signature types and origins defined by *ftSignatureType*.

Type of Journal: *ftJournalType*

The *ftJournalType* is used with the journal function and specifies the content and format of the returned journal stream. The data type is **Int64** and contains a country specific code which is a value following the ISO-3166-1-ALPHA-2 standard, converted from ASCII into hex and used as byte 8 and 7.

For definitions regarding national laws, please refer to the appropriate appendix.

Value	Description	Middleware Version
0x0000000000000000	Version information — Version information informs which version of the fiskaltrust.Middleware is currently being used.	1.1

Value	Description	Middleware Version
0x0000000000000001	fiskaltrust.ActionJournal in internal format — The fiskaltrust.ActionJournal collects all operational incidents. This can be the date and time of start or failure of the service, as well as any other information related to the fiskaltrust.Middleware and fiskaltrust.SecurityMechanism. See https://docs.fiskaltrust.eu/docs/poscreators/middleware-doc/general/cash-register-integration#fiskaltrustactionjournal .	1.1
0x0000000000000002	ReceiptJournal in internal format — The fiskaltrust.ReceiptJournal is used to record, hash, and chain all requests to the fiskaltrust.Middleware and the resulting responses. The first part of the returned ReceiptIdentification is an upcounting number generated by ReceiptJournal. See https://docs.fiskaltrust.eu/docs/poscreators/middleware-doc/general/cash-register-integration#fiskaltrustreceiptjournal .	1.1
0x0000000000000003	QueueItemJournal in internal format — QueueItemJournal shows every information related to the fiskaltrust.Middleware and fiskaltrust.SecurityMechanism, as well as all receipts sent to the fiskaltrust.Middleware and all resulting responses. This is useful for archiving purposes.	1.1

Table 8. Journal content and format options defined by *ftJournalType*.

Example for version information:

```
{
  "Assembly": "fiskaltrust.Middleware.Queue, Version=1.3.24.0, Culture=neutral, PublicKeyToken=null",
  "Version": {
    "Major": 1,
    "Minor": 3,
    "Build": 24,
    "Revision": 0,
    "MajorRevision": 0,
    "MinorRevision": 0
  }
}
```