
netsgiro documentation

Release 1.2.0

Otovo AS

Nov 22, 2018

1	License	3
2	Project resources	5
2.1	Quickstart	5
2.2	Objects API	12
2.3	Records API	16
2.4	Constants	21
2.5	Changelog	23
	Python Module Index	25

netsgiro is a Python library for working with [Nets](#) AvtaleGiro and OCR Giro files.

AvtaleGiro is a direct debit solution that is in widespread use in Norway, with more than 15 000 companies offering it to their customers. OCR Giro is used by Nets and Norwegian banks to update payees on recent deposits to their bank accounts. In combination, AvtaleGiro and OCR Giro allows for a high level of automation of invoicing and payment processing.

The netsgiro library supports:

- Parsing AvtaleGiro agreements
- Creating AvtaleGiro payment requests
- Creating AvtaleGiro cancellations
- Parsing OCR Giro transactions

netsgiro is available from PyPI. To install it, run:

```
pip install netsgiro
```

For details and code examples, see [the netsgiro documentation](#).

For further details, please refer to the official [AvtaleGiro](#) and [OCR Giro](#) documentation from Nets.

CHAPTER 1

License

Copyright 2017 [Otovo AS](#). Licensed under the Apache License, Version 2.0. See the `LICENSE` file for the full license text.

(continued from previous page)

```

Agreement (service_code=<ServiceCode.AVTALEGIRO: 21>, number=3, registration_type=
↳<AvtaleGiroRegistrationType.NEW_OR_UPDATED_AGREEMENT: 1>, kid='001020200507462',
↳notify=True),
Agreement (service_code=<ServiceCode.AVTALEGIRO: 21>, number=4, registration_type=
↳<AvtaleGiroRegistrationType.NEW_OR_UPDATED_AGREEMENT: 1>, kid='001026300507518',
↳notify=True),
Agreement (service_code=<ServiceCode.AVTALEGIRO: 21>, number=5, registration_type=
↳<AvtaleGiroRegistrationType.NEW_OR_UPDATED_AGREEMENT: 1>, kid='001044400507783',
↳notify=True),
Agreement (service_code=<ServiceCode.AVTALEGIRO: 21>, number=6, registration_type=
↳<AvtaleGiroRegistrationType.NEW_OR_UPDATED_AGREEMENT: 1>, kid='001045000507792',
↳notify=False),
Agreement (service_code=<ServiceCode.AVTALEGIRO: 21>, number=7, registration_type=
↳<AvtaleGiroRegistrationType.NEW_OR_UPDATED_AGREEMENT: 1>, kid='001057800507922',
↳notify=False),
Agreement (service_code=<ServiceCode.AVTALEGIRO: 21>, number=8, registration_type=
↳<AvtaleGiroRegistrationType.NEW_OR_UPDATED_AGREEMENT: 1>, kid='001060300509570',
↳notify=True),
Agreement (service_code=<ServiceCode.AVTALEGIRO: 21>, number=9, registration_type=
↳<AvtaleGiroRegistrationType.NEW_OR_UPDATED_AGREEMENT: 1>, kid='001087600508176',
↳notify=True),
Agreement (service_code=<ServiceCode.AVTALEGIRO: 21>, number=10, registration_type=
↳<AvtaleGiroRegistrationType.NEW_OR_UPDATED_AGREEMENT: 1>, kid='001105600508416',
↳notify=True),
Agreement (service_code=<ServiceCode.AVTALEGIRO: 21>, number=11, registration_type=
↳<AvtaleGiroRegistrationType.NEW_OR_UPDATED_AGREEMENT: 1>, kid='001123000508621',
↳notify=True),
Agreement (service_code=<ServiceCode.AVTALEGIRO: 21>, number=12, registration_type=
↳<AvtaleGiroRegistrationType.NEW_OR_UPDATED_AGREEMENT: 1>, kid='001124000508637',
↳notify=True),
Agreement (service_code=<ServiceCode.AVTALEGIRO: 21>, number=13, registration_type=
↳<AvtaleGiroRegistrationType.NEW_OR_UPDATED_AGREEMENT: 1>, kid='001138900509107',
↳notify=False),
Agreement (service_code=<ServiceCode.AVTALEGIRO: 21>, number=14, registration_type=
↳<AvtaleGiroRegistrationType.NEW_OR_UPDATED_AGREEMENT: 1>, kid='001143700509281',
↳notify=True),
Agreement (service_code=<ServiceCode.AVTALEGIRO: 21>, number=15, registration_type=
↳<AvtaleGiroRegistrationType.NEW_OR_UPDATED_AGREEMENT: 1>, kid='001146800509317',
↳notify=True),
Agreement (service_code=<ServiceCode.AVTALEGIRO: 21>, number=16, registration_type=
↳<AvtaleGiroRegistrationType.NEW_OR_UPDATED_AGREEMENT: 1>, kid='001186100509492',
↳notify=False)]

```

As you can see, all updates here are for new or updated agreements. If a payer deletes their AvtaleGiro agreement the `registration_type` field will be `DELETED_AGREEMENT`.

2.1.2 Building payment requests

Once you have established AvtaleGiro agreements with some payers, you can start creating payment requests based on your invoices.

You start by creating a *Transmission* with Nets as the recipient:

```

>>> from datetime import date
>>> from decimal import Decimal
>>> import netsgiro
>>> transmission = netsgiro.Transmission(
...     number='1703231',
...     data_transmitter='01234567',
...     data_recipient=netsgiro.NETS_ID)

```



```
>>> import netsgiro
>>> transmission = netsgiro.parse(data)
>>> transmission
Transmission(number='0170031', data_transmitter='00008080', data_recipient=
↳'00010200', date=datetime.date(1992, 1, 20))
>>> transmission.number
'0170031'
```

Each transmission can contain any number of assignments:

```
>>> len(transmission.assignments)
1
>>> assignment = transmission.assignments[0]
>>> assignment
Assignment(service_code=<ServiceCode.OCR_GIRO: 9>, type=<AssignmentType.
↳TRANSACTIONS: 0>, number='0000002', account='99991042764', agreement_id=
↳'001008566', date=datetime.date(1992, 1, 20))
>>> assignment.number
'0000002'
>>> assignment.get_earliest_transaction_date()
datetime.date(1992, 1, 20)
>>> assignment.get_latest_transaction_date()
datetime.date(1992, 1, 20)
>>> assignment.get_total_amount()
Decimal('51449')
```

Each assignment can contain any number of transactions, in this case OCR Giro payment transactions:

```
>>> pprint(assignment.transactions)
[Transaction(service_code=<ServiceCode.OCR_GIRO: 9>, type=<TransactionType.
↳PURCHASE_WITH_TEXT: 21>, number=1, date=datetime.date(1992, 1, 20),
↳amount=Decimal('1020'), kid='0000531', reference='099038562', text='Foo bar baz',
↳ centre_id='13', day_code=20, partial_settlement_number=1, partial_settlement_
↳serial_number='01464', sign='0', form_number='9636827194', bank_date=datetime.
↳date(1992, 1, 16), debit_account='99990512341', _filler='0000000'),
Transaction(service_code=<ServiceCode.OCR_GIRO: 9>, type=<TransactionType.FROM_
↳GIRO_DEBITED_ACCOUNT: 10>, number=2, date=datetime.date(1992, 1, 20),
↳amount=Decimal('1020'), kid='0036633', reference='097596016', text=None, centre_
↳id='13', day_code=20, partial_settlement_number=1, partial_settlement_serial_
↳number='01464', sign='0', form_number='9797596016', bank_date=datetime.date(1992,
↳ 1, 16), debit_account='99991005524', _filler='1883206'),
Transaction(service_code=<ServiceCode.OCR_GIRO: 9>, type=<TransactionType.FROM_
↳GIRO_DEBITED_ACCOUNT: 10>, number=3, date=datetime.date(1992, 1, 20),
↳amount=Decimal('560'), kid='0048763', reference='097581675', text=None, centre_
↳id='13', day_code=20, partial_settlement_number=1, partial_settlement_serial_
↳number='01464', sign='0', form_number='9797581675', bank_date=datetime.date(1992,
↳ 1, 17), debit_account='99991042764', _filler='1883206'),
Transaction(service_code=<ServiceCode.OCR_GIRO: 9>, type=<TransactionType.FROM_
↳GIRO_DEBITED_ACCOUNT: 10>, number=4, date=datetime.date(1992, 1, 20),
↳amount=Decimal('1020'), kid='0063851', reference='097585760', text=None, centre_
↳id='13', day_code=20, partial_settlement_number=1, partial_settlement_serial_
↳number='01464', sign='0', form_number='9797585760', bank_date=datetime.date(1992,
↳ 1, 17), debit_account='99991005524', _filler='1883206'),
Transaction(service_code=<ServiceCode.OCR_GIRO: 9>, type=<TransactionType.FROM_
↳GIRO_DEBITED_ACCOUNT: 10>, number=5, date=datetime.date(1992, 1, 20),
↳amount=Decimal('1020'), kid='0120243', reference='097591579', text=None, centre_
↳id='13', day_code=20, partial_settlement_number=1, partial_settlement_serial_
↳number='01464', sign='0', form_number='9797591579', bank_date=datetime.date(1992,
↳ 1, 17), debit_account='99990567898', _filler='1883206'),
Transaction(service_code=<ServiceCode.OCR_GIRO: 9>, type=<TransactionType.FROM_
↳GIRO_DEBITED_ACCOUNT: 10>, number=6, date=datetime.date(1992, 1, 20),
↳amount=Decimal('560'), kid='0165867', reference='097585109', text=None, centre_
↳id='13', day_code=20, partial_settlement_number=1, partial_settlement_serial_
↳number='01464', sign='0', form_number='9797585109', bank_date=datetime.date(1992,
↳ 1, 16), debit_account='99991042764', _filler='1883206'),
```

(continues on next page)

(continued from previous page)

```

Transaction(service_code=<ServiceCode.OCR_GIRO: 9>, type=<TransactionType.FROM_
↳GIRO_DEBITED_ACCOUNT: 10>, number=7, date=datetime.date(1992, 1, 20),
↳amount=Decimal('1020'), kid='0178357', reference='097584855', text=None, centre_
↳id='13', day_code=20, partial_settlement_number=1, partial_settlement_serial_
↳number='01464', sign='0', form_number='9797584855', bank_date=datetime.date(1992,
↳ 1, 16), debit_account='99991005524', _filler='1883206'),
Transaction(service_code=<ServiceCode.OCR_GIRO: 9>, type=<TransactionType.FROM_
↳STANDING_ORDERS: 11>, number=8, date=datetime.date(1992, 1, 20), amount=Decimal(
↳'1500'), kid='02212291038306', reference='600155380', text=None, centre_id='13',
↳day_code=20, partial_settlement_number=1, partial_settlement_serial_number='01570
↳', sign='0', form_number='9600000000', bank_date=datetime.date(1992, 1, 20),
↳debit_account='99999545528', _filler='0000000'),
Transaction(service_code=<ServiceCode.OCR_GIRO: 9>, type=<TransactionType.FROM_
↳DIRECT_REMITTANCE: 12>, number=9, date=datetime.date(1992, 1, 20),
↳amount=Decimal('1200'), kid='02311291038304', reference='001020169', text=None,
↳centre_id='13', day_code=20, partial_settlement_number=1, partial_settlement_
↳serial_number='01570', sign='0', form_number='0000000000', bank_date=None, debit_
↳account='99991015406', _filler='0000000'),
Transaction(service_code=<ServiceCode.OCR_GIRO: 9>, type=<TransactionType.FROM_
↳BUSINESS_TERMINAL_GIRO: 13>, number=10, date=datetime.date(1992, 1, 20),
↳amount=Decimal('550'), kid='02310291038308', reference='220663800', text=None,
↳centre_id='13', day_code=20, partial_settlement_number=1, partial_settlement_
↳serial_number='01570', sign='0', form_number='0000000000', bank_date=None, debit_
↳account='99991017530', _filler='0000000'),
...
Transaction(service_code=<ServiceCode.OCR_GIRO: 9>, type=<TransactionType.FROM_
↳GIRO_DEBITED_ACCOUNT: 10>, number=20, date=datetime.date(1992, 1, 20),
↳amount=Decimal('540'), kid='02311291133188', reference='091308861', text=None,
↳centre_id='13', day_code=20, partial_settlement_number=9, partial_settlement_
↳serial_number='04514', sign='0', form_number='9636778117', bank_date=datetime.
↳date(1992, 1, 17), debit_account='99991011125', _filler='0000000')]

```

2.2 Objects API

The objects API is the higher level API. It parses the “OCR” file into an object structure that for most use cases is more pleasant to work with than the lower level *records API*.

See *the quickstart guide* for an introduction to how to use the API.

```

netsgiro.parse(data: str) → netsgiro.objects.Transmission
    Parse an OCR file into a Transmission object.

```

2.2.1 Transmission class

```

class netsgiro.Transmission(number, data_transmitter, data_recipient, date=None, assign-
    ments=NOTHING)

```

Transmission is the top-level object.

An OCR file contains a single transmission. The transmission can contain multiple *Assignment* objects of various types.

```

add_assignment(*, service_code: netsgiro.enums.ServiceCode, assignment_type: nets-
    giro.enums.AssignmentType, agreement_id: Optional[str] = None, num-
    ber: str, account: str, date: Optional[datetime.date] = None) → nets-
    giro.objects.Assignment

```

Add an assignment to the transmission.

```

classmethod from_records(records: List[netsgiro.records.Record]) → nets-
    giro.objects.Transmission

```

Build a Transmission object from a list of record objects.

get_num_records () → int
Get number of records in the transmission.
Includes the transmission's start and end record.

get_num_transactions () → int
Get number of transactions in the transmission.

get_total_amount () → decimal.Decimal
Get the total amount from all transactions in the transmission.

to_ocr () → str
Convert the transmission to an OCR string.

to_records () → Iterable[netsgiro.records.Record]
Convert the transmission to a list of records.

assignments = None
List of assignments.

data_recipient = None
Data recipient's Nets ID. String of 8 digits.

data_transmitter = None
Data transmitter's Nets ID. String of 8 digits.

date = None
For OCR Giro files from Nets, this is Nets' processing date.
For AvtaleGiro payment request, the earliest due date in the transmission is automatically used.

number = None
Data transmitters unique enumeration of the transmission. String of 7 digits.

2.2.2 Assignment class

class netsgiro.**Assignment** (*service_code, type, number, account, agreement_id=None, date=None, transactions=NOTHING*)
An Assignment groups multiple transactions within a transmission.

Use `netsgiro.Transmission.add_assignment()` to create assignments.

add_payment_cancellation (*, *kid: str, due_date: datetime.date, amount: decimal.Decimal, reference: Optional[str] = None, payer_name: Optional[str] = None, bank_notification: Union[bool, str] = False*) → netsgiro.objects.Transaction

Add an AvtaleGiro cancellation to the assignment.

The assignment must have service code `AVTALEGIRO` and assignment type `AVTALEGIRO_CANCELLATIONS`.

Otherwise, the cancellation must be identical to the payment request it is cancelling.

add_payment_request (*, *kid: str, due_date: datetime.date, amount: decimal.Decimal, reference: Optional[str] = None, payer_name: Optional[str] = None, bank_notification: Union[bool, str] = False*) → netsgiro.objects.Transaction

Add an AvtaleGiro payment request to the assignment.

The assignment must have service code `AVTALEGIRO` and assignment type `TRANSACTIONS`.

classmethod from_records (*records: List[netsgiro.records.Record]*) → netsgiro.objects.Assignment

Build an Assignment object from a list of record objects.

get_earliest_transaction_date () → Optional[datetime.date]

Get earliest date from the assignment's transactions.

get_latest_transaction_date () → Optional[datetime.date]
 Get latest date from the assignment's transactions.

get_num_records () → int
 Get number of records in the assignment.
 Includes the assignment's start and end record.

get_num_transactions () → int
 Get number of transactions in the assignment.

get_total_amount () → decimal.Decimal
 Get the total amount from all transactions in the assignment.

to_records () → Iterable[netsgiro.records.Record]
 Convert the assignment to a list of records.

account = None
 The payee's bank account. String of 11 digits.

agreement_id = None
 Used for OCR Giro.
 The payee's agreement ID with Nets. String of 9 digits.

date = None
 Used for OCR Giro.
 The date the assignment was generated by Nets.

number = None
 The assignment number. String of 7 digits.

service_code = None
 The service code. One of *ServiceCode*.

transactions = None
 List of transaction objects, like *Agreement*, *PaymentRequest*, *Transaction*.

type = None
 The transaction type. One of *TransactionType*.

2.2.3 Transaction classes

class netsgiro.**Agreement** (*service_code*, *number*, *registration_type*, *kid*, *notify*)
 Agreement contains an AvtaleGiro agreement update.

Agreements are only found in assignments of the *AVTALEGIRO_AGREEMENTS* type, which are only created by Nets.

classmethod **from_records** (*records*: List[netsgiro.records.Record]) → nets-
 giro.objects.Agreement
 Build an Agreement object from a list of record objects.

to_records () → Iterable[netsgiro.records.Record]
 Convert the agreement to a list of records.

TRANSACTION_TYPE = 94

kid = None
 KID number to identify the customer and invoice.

notify = None
 Whether the payer wants notification about payment requests.

number = None
 Transaction number. Unique and ordered within an assignment.

registration_type = None

Type of agreement registration update. One of *AvtaleGiroRegistrationType*.

service_code = None

The service code. One of *ServiceCode*.

class netsgiro.**PaymentRequest** (*service_code, type, number, date, amount, kid, reference, text, payer_name*)

PaymentRequest contains an AvtaleGiro payment request or cancellation.

To create a transaction, you will normally use the helper methods on *Assignment*: *add_payment_request()* and *add_payment_cancellation()*.

classmethod **from_records** (*records: List[netsgiro.records.Record]*) → netsgiro.objects.Transaction

Build a Transaction object from a list of record objects.

to_records () → Iterable[netsgiro.records.Record]

Convert the transaction to a list of records.

amount = None

Transaction amount in NOK with two decimals.

amount_in_cents

Transaction amount in NOK cents.

date = None

The due date.

kid = None

KID number to identify the customer and invoice.

number = None

Transaction number. Unique and ordered within an assignment.

payer_name = None

The value is only used to help the payee cross-reference reports from Nets with their own records. It is not visible to the payer.

reference = None

This is a specification line that will, if set, be displayed on the payers account statement. Alphanumeric, max 25 chars.

service_code = None

The service code. One of *ServiceCode*.

text = None

This is up to 42 lines of 80 chars each of free text used by the bank to notify the payer about the payment request. It is not used if the payee is responsible for notifying the payer.

type = None

The transaction type. One of *TransactionType*.

class netsgiro.**Transaction** (*service_code, type, number, date, amount, kid, reference, text, centre_id, day_code, partial_settlement_number, partial_settlement_serial_number, sign, form_number, bank_date, debit_account, filler*)

Transaction contains an OCR Giro transaction.

Transactions are found in assignments with the service code *OCR_GIRO* type, which are only created by Nets.

classmethod **from_records** (*records: List[netsgiro.records.Record]*) → netsgiro.objects.Transaction

Build a Transaction object from a list of record objects.

to_records () → Iterable[netsgiro.records.Record]

Convert the transaction to a list of records.

amount = None
Transaction amount in NOK with two decimals.

amount_in_cents
Transaction amount in NOK cents.

bank_date = None
Used for OCR Giro.

centre_id = None
Used for OCR Giro.

date = None
Nets' processing date.

day_code = None
Used for OCR Giro.

debit_account = None
Used for OCR Giro.

form_number = None
Used for OCR Giro.

kid = None
KID number to identify the customer and invoice.

number = None
Transaction number. Unique and ordered within an assignment.

partial_settlement_number = None
Used for OCR Giro.

partial_settlement_serial_number = None
Used for OCR Giro.

reference = None
The value depends on the payment method.

service_code = None
The service code. One of *ServiceCode*.

sign = None
Used for OCR Giro.

text = None
Up to 40 chars of free text from the payment terminal.

type = None
The transaction type. One of *TransactionType*.

2.3 Records API

The records API is the lower level API. It parses each line of “OCR” text input into a record object. A record object also knows about its OCR representation.

For most use cases, the *objects API* is preferable to the records API.

2.3.1 File parsing

To parse an OCR file, you must first read the contents of the OCR file. It should be decoded using the ISO-8559-1 encoding:

Each transmission can contain any number of assignments.

classmethod from_string (*line: str*) → netsgiro.records.Record
Parse OCR string into a record object.

to_ocr () → str
Get record as OCR string.

RECORD_TYPE = 10

class netsgiro.records.**TransmissionEnd** (*service_code, num_transactions, num_records, total_amount, nets_date*)

TransmissionEnd is the first record in every OCR file.

classmethod from_string (*line: str*) → netsgiro.records.Record
Parse OCR string into a record object.

to_ocr () → str
Get record as OCR string.

RECORD_TYPE = 89

class netsgiro.records.**AssignmentStart** (*service_code, assignment_type, assignment_number, assignment_account, agreement_id=None*)

AssignmentStart is the first record of an assignment.

Each assignment can contain any number of transactions.

classmethod from_string (*line: str*) → netsgiro.records.Record
Parse OCR string into a record object.

to_ocr () → str
Get record as OCR string.

RECORD_TYPE = 20

class netsgiro.records.**AssignmentEnd** (*service_code, assignment_type, num_transactions, num_records, total_amount=None, nets_date_1=None, nets_date_2=None, nets_date_3=None*)

AssignmentEnd is the last record of an assignment.

classmethod from_string (*line: str*) → netsgiro.records.Record
Parse OCR string into a record object.

to_ocr () → str
Get record as OCR string.

RECORD_TYPE = 88

nets_date
Nets' processing date.

Only used for OCR Giro.

nets_date_earliest
Earliest date from the contained transactions.

nets_date_latest
Latest date from the contained transactions.

class netsgiro.records.**TransactionAmountItem1** (*service_code, transaction_type, transaction_number, nets_date, amount, kid, centre_id=None, day_code=None, partial_settlement_number=None, partial_settlement_serial_number=None, sign=None*)

TransactionAmountItem1 is the first record of a transaction.

The record is used both for AvtaleGiro and for OCR Giro.

classmethod from_string (*line: str*) → netsgiro.records.Record
 Parse OCR string into a record object.

to_ocr () → str
 Get record as OCR string.

RECORD_TYPE = 30

```
class netsgiro.records.TransactionAmountItem2 (service_code, transaction_type,
                                              transaction_number, reference,
                                              form_number=None,
                                              bank_date=None,
                                              debit_account=None, filler=None,
                                              payer_name=None)
```

TransactionAmountItem2 is the second record of a transaction.

The record is used both for AvtaleGiro and for OCR Giro.

classmethod from_string (*line: str*) → netsgiro.records.Record
 Parse OCR string into a record object.

to_ocr () → str
 Get record as OCR string.

RECORD_TYPE = 31

```
class netsgiro.records.TransactionAmountItem3 (service_code, transaction_type,
                                              transaction_number, text)
```

TransactionAmountItem3 is the third record of a transaction.

The record is only used for some OCR Giro transaction types.

classmethod from_string (*line: str*) → netsgiro.records.Record
 Parse OCR string into a record object.

to_ocr () → str
 Get record as OCR string.

RECORD_TYPE = 32

```
class netsgiro.records.TransactionSpecification (service_code, transaction_type,
                                              transaction_number, line_number,
                                              column_number, text)
```

TransactionSpecification is used for AvtaleGiro transactions.

The record is only used when bank notification is used to notify the payer.

Each record contains half of an 80 char long line of text and can be repeated up to 84 times for a single transaction for a total of 42 lines of specification text.

classmethod from_string (*line: str*) → netsgiro.records.Record
 Parse OCR string into a record object.

classmethod from_text (*, *service_code, transaction_type, transaction_number, text*) → Iterable[TransactionSpecification]
 Create a sequence of specification records from a text string.

to_ocr () → str
 Get record as OCR string.

classmethod to_text (*records: Sequence[TransactionSpecification]*) → str
 Get a text string from a sequence of specification records.

RECORD_TYPE = 49

class netsgiro.records.**AvtaleGiroAgreement** (*service_code, transaction_type, transaction_number, registration_type, kid, notify*)

AvtaleGiroAgreement is used by Nets to notify about agreement changes.

This includes new or deleted agreements, as well as updates to the payer's notification preferences.

classmethod **from_string** (*line: str*) → netsgiro.records.Record
Parse OCR string into a record object.

to_ocr () → str
Get record as OCR string.

RECORD_TYPE = 70

2.4 Constants

netsgiro.**NETS_ID**
Nets' data transmitter/recipient ID

```
>>> import netsgiro
>>> netsgiro.NETS_ID
'00008080'
```

class netsgiro.**ServiceCode**
Service codes tell which Nets service the record applies to.

NONE = 0
Used for the transmission start and end record.

OCR_GIRO = 9
Used for all OCR Giro records.

AVTALEGIRO = 21
Used for all AvtaleGiro records.

class netsgiro.**RecordType**
Record types tell what type of record this is.

TRANSMISSION_START = 10
See netsgiro.TransmissionStart.

ASSIGNMENT_START = 20
See netsgiro.AssignmentStart.

TRANSACTION_AMOUNT_ITEM_1 = 30
See netsgiro.TransactionAmountItem1.

TRANSACTION_AMOUNT_ITEM_2 = 31
See netsgiro.TransactionAmountItem2.

TRANSACTION_AMOUNT_ITEM_3 = 32
See netsgiro.TransactionAmountItem3.

TRANSACTION_SPECIFICATION = 49
See netsgiro.TransactionSpecification.

TRANSACTION_AGREEMENTS = 70
See netsgiro.TransactionAgreements.

ASSIGNMENT_END = 88
See netsgiro.AssignmentEnd.

TRANSMISSION_END = 89
See netsgiro.TransmissionEnd.

class netsgiro.**AssignmentType**

Assignment types tell what type of assignment this is.

TRANSACTIONS = 0

Used both for AvtaleGiro payment requests and OCR Giro transactions.

AVTALEGIRO_AGREEMENTS = 24

Used for AvtaleGiro agreement updates.

AVTALEGIRO_CANCELLATIONS = 36

Used for AvtaleGiro cancellations.

class netsgiro.**TransactionType**

Assignment types tell what type of transaction this is.

FROM_GIRO_DEBITED_ACCOUNT = 10

Used for OCR Giro.

FROM_STANDING_ORDERS = 11

Used for OCR Giro.

FROM_DIRECT_REMITTANCE = 12

Used for OCR Giro.

FROM_BUSINESS_TERMINAL_GIRO = 13

Used for OCR Giro.

FROM_COUNTER_GIRO = 14

Used for OCR Giro.

FROM_AVTALEGIRO = 15

Used for OCR Giro.

FROM_TELEGIRO = 16

Used for OCR Giro.

FROM_CASH_GIRO = 17

Used for OCR Giro.

REVERSING_WITH_KID = 18

Used for OCR Giro.

PURCHASE_WITH_KID = 19

Used for OCR Giro.

REVERSING_WITH_TEXT = 20

Used for OCR Giro.

PURCHASE_WITH_TEXT = 21

Used for OCR Giro.

Note: The value 21 is used for both *PURCHASE_WITH_TEXT* and *AVTALEGIRO_WITH_BANK_NOTIFICATION*. The enum representation will be `<PURCHASE_WITH_TEXT: 21>` in either case.

AVTALEGIRO_WITH_PAYEE_NOTIFICATION = 2

Used for AvtaleGiro when you want to notify the payer yourself.

AVTALEGIRO_WITH_BANK_NOTIFICATION = 21

Used for AvtaleGiro when you want the bank to notify the payer.

AVTALEGIRO_CANCELLATION = 93

Used for transactions that are part of an AvtaleGiro cancellation assignment.

AVTALEGIRO_AGREEMENT = 94

Used by Nets for updates to AvtaleGiro agreements.

class `netsgiro.AvtaleGiroRegistrationType`

AvtaleGiro registration types tell what kind of update this is.

ACTIVE_AGREEMENT = 0

Used when the AvtaleGiro agreement assignment contains all currently active agreements.

NEW_OR_UPDATED_AGREEMENT = 1

Used when the AvtaleGiro agreement assignment contains only changes, and the current agreement is new or updated.

DELETED_AGREEMENT = 2

Used when the AvtaleGiro agreement assignment contains only changes, and the current agreement has been deleted.

2.5 Changelog

2.5.1 v1.2.0 (2018-11-22)

- Python 3.7 is now explicitly supported. No changes was required to the library, but the test suite now runs on Python 3.4, 3.5, 3.6, and 3.7.
- Require `attrs >= 17.4`, and replace all use of the deprecated keyword argument `convert` with `converter` so that `netsgiro` can continue working with `attrs` releases after January 2019.
- Format code with Black.

2.5.2 v1.1.0 (2017-10-23)

Record parsing and writing are now tested with an additional suite of property based tests, using the Hypothesis library. This testing effort identified a number of issues, all of which are fixed by this release.

- Fixed exception in `netsgiro.records.TransactionAmountItem2.to_ocr()` if `payer_name` was `None`.
- Fixed exception in `netsgiro.records.TransactionAmountItem3.to_ocr()` if `text` was `None`.
- Raise a `ValueError` if a too long string is used for any of:
 - `netsgiro.records.TransactionAmountItem1.kid` (max 25 chars)
 - `netsgiro.records.TransactionAmountItem3.text` (max 40 chars)
 - `netsgiro.records.TransactionSpecification.text` (max 40 chars)
 - `netsgiro.records.AvtaleGiroAgreement.kid` (max 25 chars)

Previously the string was accepted and the record generated invalid OCR data.

- Strip newline characters (`\n` and `\r`) from record strings, like `netsgiro.records.TransactionAmountItem2.payer_name`.

Previously the newline characters were accepted and the record generated invalid OCR data.

- Automatically pad `netsgiro.records.TransactionSpecification.text` to 40 chars, so that a manually created record and a record parsed from OCR are identical.

2.5.3 v1.0.0 (2017-05-20)

No changes from v1.0.0a3, which has been used in production for a few weeks without any issues.

2.5.4 v1.0.0a3 (2017-05-03)

- Rename `netsgiro.TransactionType.AVTALEGIRO_AGREEMENTS` (plural) to `netsgiro.TransactionType.AVTALEGIRO_AGREEMENT` (singular).
- When writing record to OCR, cut `netsgiro.records.TransactionAmountItem2.payer_name` to 10 first chars, as that is all the field has room for.

2.5.5 v1.0.0a2 (2017-04-26)

Major improvements and changes.

- The objects API now supports parsing all known file variants with `netsgiro.parse()` and can recreate the parsed OCR data perfectly with `netsgiro.Transmission.to_ocr()`.
- The objects API now does all the bookkeeping necessary for building payment requests. With this improvement, the code necessary to produce a payment request is cut to from around 100 to 25 lines of code.
- New *quickstart guide* shows how to parse files and build payment requests.
- All public methods and fields of both the *objects API* and *records API* are now documented.
- The low-level *records API* has been moved to the `netsgiro.records` module.

2.5.6 v1.0.0a1 (2017-04-17)

Initial alpha release. No promises about backwards compatibility.

n

`netsgiro`, 12

`netsgiro.records`, 16

A

account (netsgiro.Assignment attribute), 14
 ACTIVE_AGREEMENT (netsgiro.AvtaleGiroRegistrationType attribute), 23
 add_assignment() (netsgiro.Transmission method), 12
 add_payment_cancellation() (netsgiro.Assignment method), 13
 add_payment_request() (netsgiro.Assignment method), 13
 Agreement (class in netsgiro), 14
 agreement_id (netsgiro.Assignment attribute), 14
 amount (netsgiro.PaymentRequest attribute), 15
 amount (netsgiro.Transaction attribute), 15
 amount_in_cents (netsgiro.PaymentRequest attribute), 15
 amount_in_cents (netsgiro.Transaction attribute), 16
 Assignment (class in netsgiro), 13
 ASSIGNMENT_END (netsgiro.RecordType attribute), 21
 ASSIGNMENT_START (netsgiro.RecordType attribute), 21
 AssignmentEnd (class in netsgiro.records), 19
 assignments (netsgiro.Transmission attribute), 13
 AssignmentStart (class in netsgiro.records), 19
 AssignmentType (class in netsgiro), 21
 AVTALEGIRO (netsgiro.ServiceCode attribute), 21
 AVTALEGIRO_AGREEMENT (netsgiro.TransactionType attribute), 22
 AVTALEGIRO_AGREEMENTS (netsgiro.AssignmentType attribute), 22
 AVTALEGIRO_CANCELLATION (netsgiro.TransactionType attribute), 22
 AVTALEGIRO_CANCELLATIONS (netsgiro.AssignmentType attribute), 22
 AVTALEGIRO_WITH_BANK_NOTIFICATION (netsgiro.TransactionType attribute), 22
 AVTALEGIRO_WITH_PAYEE_NOTIFICATION (netsgiro.TransactionType attribute), 22
 AvtaleGiroAgreement (class in netsgiro.records), 20
 AvtaleGiroRegistrationType (class in netsgiro), 22

B

bank_date (netsgiro.Transaction attribute), 16

C

centre_id (netsgiro.Transaction attribute), 16

D

data_recipient (netsgiro.Transmission attribute), 13
 data_transmitter (netsgiro.Transmission attribute), 13
 date (netsgiro.Assignment attribute), 14
 date (netsgiro.PaymentRequest attribute), 15
 date (netsgiro.Transaction attribute), 16
 date (netsgiro.Transmission attribute), 13
 day_code (netsgiro.Transaction attribute), 16
 debit_account (netsgiro.Transaction attribute), 16
 DELETED_AGREEMENT (netsgiro.AvtaleGiroRegistrationType attribute), 23

F

form_number (netsgiro.Transaction attribute), 16
 FROM_AVTALEGIRO (netsgiro.TransactionType attribute), 22
 FROM_BUSINESS_TERMINAL_GIRO (netsgiro.TransactionType attribute), 22
 FROM_CASH_GIRO (netsgiro.TransactionType attribute), 22
 FROM_COUNTER_GIRO (netsgiro.TransactionType attribute), 22
 FROM_DIRECT_REMITTANCE (netsgiro.TransactionType attribute), 22
 FROM_GIRO_DEBITED_ACCOUNT (netsgiro.TransactionType attribute), 22
 from_records() (netsgiro.Agreement class method), 14
 from_records() (netsgiro.Assignment class method), 13
 from_records() (netsgiro.PaymentRequest class method), 15
 from_records() (netsgiro.Transaction class method), 15
 from_records() (netsgiro.Transmission class method), 12
 FROM_STANDING_ORDERS (netsgiro.TransactionType attribute), 22
 from_string() (netsgiro.records.AssignmentEnd class method), 19
 from_string() (netsgiro.records.AssignmentStart class method), 19

from_string() (netsgiro.records.AvtaleGiroAgreement class method), 21
 from_string() (netsgiro.records.TransactionAmountItem1 class method), 20
 from_string() (netsgiro.records.TransactionAmountItem2 class method), 20
 from_string() (netsgiro.records.TransactionAmountItem3 class method), 20
 from_string() (netsgiro.records.TransactionSpecification class method), 20
 from_string() (netsgiro.records.TransmissionEnd class method), 19
 from_string() (netsgiro.records.TransmissionStart class method), 19
 FROM_TELEGIRO (netsgiro.TransactionType attribute), 22
 from_text() (netsgiro.records.TransactionSpecification class method), 20

G

get_earliest_transaction_date() (netsgiro.Assignment method), 13
 get_latest_transaction_date() (netsgiro.Assignment method), 13
 get_num_records() (netsgiro.Assignment method), 14
 get_num_records() (netsgiro.Transmission method), 13
 get_num_transactions() (netsgiro.Assignment method), 14
 get_num_transactions() (netsgiro.Transmission method), 13
 get_total_amount() (netsgiro.Assignment method), 14
 get_total_amount() (netsgiro.Transmission method), 13

K

kid (netsgiro.Agreement attribute), 14
 kid (netsgiro.PaymentRequest attribute), 15
 kid (netsgiro.Transaction attribute), 16

N

nets_date (netsgiro.records.AssignmentEnd attribute), 19
 nets_date_earliest (netsgiro.records.AssignmentEnd attribute), 19
 nets_date_latest (netsgiro.records.AssignmentEnd attribute), 19
 netsgiro (module), 12
 netsgiro.NETS_ID (built-in variable), 21
 netsgiro.records (module), 16
 NEW_OR_UPDATED_AGREEMENT (netsgiro.AvtaleGiroRegistrationType attribute), 23
 NONE (netsgiro.ServiceCode attribute), 21
 notify (netsgiro.Agreement attribute), 14
 number (netsgiro.Agreement attribute), 14
 number (netsgiro.Assignment attribute), 14
 number (netsgiro.PaymentRequest attribute), 15
 number (netsgiro.Transaction attribute), 16
 number (netsgiro.Transmission attribute), 13

O

OCR_GIRO (netsgiro.ServiceCode attribute), 21

P

parse() (in module netsgiro), 12
 parse() (in module netsgiro.records), 18
 partial_settlement_number (netsgiro.Transaction attribute), 16
 partial_settlement_serial_number (netsgiro.Transaction attribute), 16
 payer_name (netsgiro.PaymentRequest attribute), 15
 PaymentRequest (class in netsgiro), 15
 PURCHASE_WITH_KID (netsgiro.TransactionType attribute), 22
 PURCHASE_WITH_TEXT (netsgiro.TransactionType attribute), 22

R

RECORD_TYPE (netsgiro.records.AssignmentEnd attribute), 19
 RECORD_TYPE (netsgiro.records.AssignmentStart attribute), 19
 RECORD_TYPE (netsgiro.records.AvtaleGiroAgreement attribute), 21
 RECORD_TYPE (netsgiro.records.TransactionAmountItem1 attribute), 20
 RECORD_TYPE (netsgiro.records.TransactionAmountItem2 attribute), 20
 RECORD_TYPE (netsgiro.records.TransactionAmountItem3 attribute), 20
 RECORD_TYPE (netsgiro.records.TransactionSpecification attribute), 20
 RECORD_TYPE (netsgiro.records.TransmissionEnd attribute), 19
 RECORD_TYPE (netsgiro.records.TransmissionStart attribute), 19
 RecordType (class in netsgiro), 21
 reference (netsgiro.PaymentRequest attribute), 15
 reference (netsgiro.Transaction attribute), 16
 registration_type (netsgiro.Agreement attribute), 14
 REVERSING_WITH_KID (netsgiro.TransactionType attribute), 22
 REVERSING_WITH_TEXT (netsgiro.TransactionType attribute), 22

S

service_code (netsgiro.Agreement attribute), 15
 service_code (netsgiro.Assignment attribute), 14
 service_code (netsgiro.PaymentRequest attribute), 15
 service_code (netsgiro.Transaction attribute), 16
 ServiceCode (class in netsgiro), 21
 sign (netsgiro.Transaction attribute), 16

T

- text (netsgiro.PaymentRequest attribute), 15
- text (netsgiro.Transaction attribute), 16
- to_ocr() (netsgiro.records.AssignmentEnd method), 19
- to_ocr() (netsgiro.records.AssignmentStart method), 19
- to_ocr() (netsgiro.records.AvtaleGiroAgreement method), 21
- to_ocr() (netsgiro.records.TransactionAmountItem1 method), 20
- to_ocr() (netsgiro.records.TransactionAmountItem2 method), 20
- to_ocr() (netsgiro.records.TransactionAmountItem3 method), 20
- to_ocr() (netsgiro.records.TransactionSpecification method), 20
- to_ocr() (netsgiro.records.TransmissionEnd method), 19
- to_ocr() (netsgiro.records.TransmissionStart method), 19
- to_ocr() (netsgiro.Transmission method), 13
- to_records() (netsgiro.Agreement method), 14
- to_records() (netsgiro.Assignment method), 14
- to_records() (netsgiro.PaymentRequest method), 15
- to_records() (netsgiro.Transaction method), 15
- to_records() (netsgiro.Transmission method), 13
- to_text() (netsgiro.records.TransactionSpecification class method), 20
- Transaction (class in netsgiro), 15
- TRANSACTION_AGREEMENTS (netsgiro.RecordType attribute), 21
- TRANSACTION_AMOUNT_ITEM_1 (netsgiro.RecordType attribute), 21
- TRANSACTION_AMOUNT_ITEM_2 (netsgiro.RecordType attribute), 21
- TRANSACTION_AMOUNT_ITEM_3 (netsgiro.RecordType attribute), 21
- TRANSACTION_SPECIFICATION (netsgiro.RecordType attribute), 21
- TRANSACTION_TYPE (netsgiro.Agreement attribute), 14
- TransactionAmountItem1 (class in netsgiro.records), 19
- TransactionAmountItem2 (class in netsgiro.records), 20
- TransactionAmountItem3 (class in netsgiro.records), 20
- transactions (netsgiro.Assignment attribute), 14
- TRANSACTIONS (netsgiro.AssignmentType attribute), 22
- TransactionSpecification (class in netsgiro.records), 20
- TransactionType (class in netsgiro), 22
- Transmission (class in netsgiro), 12
- TRANSMISSION_END (netsgiro.RecordType attribute), 21
- TRANSMISSION_START (netsgiro.RecordType attribute), 21
- TransmissionEnd (class in netsgiro.records), 19
- TransmissionStart (class in netsgiro.records), 18
- type (netsgiro.Assignment attribute), 14
- type (netsgiro.PaymentRequest attribute), 15
- type (netsgiro.Transaction attribute), 16