

---

# **netsgiro documentation**

*Release 1.3.0*

**Otovo AS**

**Aug 25, 2022**



|                            |           |
|----------------------------|-----------|
| <b>1 License</b>           | <b>3</b>  |
| <b>2 Project resources</b> | <b>5</b>  |
| 2.1 Quickstart . . . . .   | 5         |
| 2.2 Objects API . . . . .  | 12        |
| 2.3 Records API . . . . .  | 17        |
| 2.4 Constants . . . . .    | 23        |
| 2.5 Changelog . . . . .    | 25        |
| <b>Python Module Index</b> | <b>29</b> |
| <b>Index</b>               | <b>31</b> |



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 2017-2019 [Otovo AS](#). Licensed under the Apache License, Version 2.0. See the `LICENSE` file for the full license text.







(continued from previous page)

```

... NY21947000000061      ↵
↵001045000507792N00000000000000000000000000000000000000000000000000000000
... NY21947000000071      ↵
↵001057800507922N00000000000000000000000000000000000000000000000000000000
... NY21947000000081      ↵
↵001060300509570J00000000000000000000000000000000000000000000000000000000
... NY21947000000091      ↵
↵001087600508176J00000000000000000000000000000000000000000000000000000000
... NY21947000000101      ↵
↵001105600508416J00000000000000000000000000000000000000000000000000000000
... NY21947000000111      ↵
↵001123000508621J00000000000000000000000000000000000000000000000000000000
... NY21947000000121      ↵
↵001124000508637J00000000000000000000000000000000000000000000000000000000
... NY21947000000131      ↵
↵001138900509107N00000000000000000000000000000000000000000000000000000000
... NY21947000000141      ↵
↵001143700509281J00000000000000000000000000000000000000000000000000000000
... NY21947000000151      ↵
↵001146800509317J00000000000000000000000000000000000000000000000000000000
... NY21947000000161      ↵
↵001186100509492N00000000000000000000000000000000000000000000000000000000
... ↵
↵NY2124880000001600000018000000000000000000000000000000000000000000000000000000
... ↵
↵NY0000890000001600000020000000000000000000000019041700000000000000000000000000
... "".strip()

```

`netsgiro.parse()` parses the input and returns a `netsgiro.Transmission` object:

```

>>> import netsgiro
>>> transmission = netsgiro.parse(data)
>>> transmission
Transmission(number='1091949', data_transmitter='00008080', data_recipient=
↵'00010200', date=datetime.date(2017, 4, 19))
>>> transmission.number
'1091949'

```

Each transmission can contain any number of assignments:

```

>>> len(transmission.assignments)
1
>>> assignment = transmission.assignments[0]
>>> assignment
Assignment(service_code=<ServiceCode.AVTALEGIRO: 21>, type=<AssignmentType.
↵AVTALEGIRO_AGREEMENTS: 24>, number='0000002', account='99991042764', agreement_
↵id=None, date=None)
>>> assignment.number
'0000002'

```

Each assignment can contain any number of transactions, in this case AvtaleGiro agreement updates:

```

>>> pprint(assignment.transactions)
[Agreement(service_code=<ServiceCode.AVTALEGIRO: 21>, number=1, registration_type=
↵<AvtaleGiroRegistrationType.NEW_OR_UPDATED_AGREEMENT: 1>, kid='000112000507155', ↵
↵notify=True),
  Agreement(service_code=<ServiceCode.AVTALEGIRO: 21>, number=2, registration_type=
↵<AvtaleGiroRegistrationType.NEW_OR_UPDATED_AGREEMENT: 1>, kid='001006300507304', ↵
↵notify=False),
  Agreement(service_code=<ServiceCode.AVTALEGIRO: 21>, number=3, registration_type=
↵<AvtaleGiroRegistrationType.NEW_OR_UPDATED_AGREEMENT: 1>, kid='001020200507462', ↵
↵notify=True),

```

(continues on next page)

(continued from previous page)

```

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)

```

Then, add an AvtaleGiro transaction assignment to the transmission using `add_assignment()`:





(continued from previous page)

```

...
↳NY0910310000007979758485509758485518832061601929999100552400000000000000000000
... NY091130000000820019213201015700000000000000150000
↳02212291038306000000
...
↳NY0911310000008960000000060015538000000002001929999954552800000000000000000000
... NY091230000000920019213201015700000000000000120000
↳02311291038304000000
...
↳NY0912310000009000000000000102016900000000000099991015406000000000000000000000
... NY09133000000102001921320101570000000000000055000
↳02310291038308000000
...
↳NY0913310000010000000000022066380000000000000099991017530000000000000000000000
... NY091130000001120019213202315700000000000000194300
↳000149012000000
...
↳NY09113100000119600000000600155211000000020019299990508034000000000000000000000
... NY091230000001220019213202315700000000000000100000
↳01211291038306000000
...
↳NY0912310000012000000000000102016900000000000999905080340000000000000000000000
... NY09133000000132001921320231570000000000000050000
↳02111291038305000000
...
↳NY0913310000013000000000003206637000000000000099990523084000000000000000000000
... NY091330000001420019213203355720000000000002050000
↳02321291038303000000
...
↳NY0913310000014000000000002506637000000000000999905100550000000000000000000000
... NY09133000000152001921320941570000000000000550000
↳02331291038302000000
...
↳NY0913310000015000000000079748962600000000000999910111250000000000000000000000
... NY09133000000162001921320941570000000000001050000
↳02341291038301000000
...
↳NY0913310000016000000000002811637000000000000999995455280000000000000000000000
... NY0910300000017200192132094157000000000000064400
↳02358291038305000000
...
↳NY0910310000017000000000028066380000000000009999954552800000000000000000000000
... NY0910300000018200192132090451400000000000056400
↳02311291029238000000
...
↳NY0910310000018963677800309130671400000017019299991015406000000000000000000000
... NY0910300000019200192132090451400000000000028800
↳02311291034832000000
...
↳NY0910310000019963677821097694999000000016019299991008034000000000000000000000
... NY0910300000020200192132090451400000000000054000
↳02311291133188000000
...
↳NY0910310000020963677811709130886100000017019299991011125000000000000000000000
...
↳NY090088000000200000004300000000005144900200192200192200192000000000000000000000
...
↳NY0000890000002000000045000000000051449002001920000000000000000000000000000000
... """".strip()

```

`netsgiro.parse()` parses the input and returns a `netsgiro.Transmission` object:

```
>>> 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: str, data_transmitter: str, data_recipient: str, date: Op-
    tional[datetime.date] = None, assignments: List[Assignment] =
    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[R]*) → netsgiro.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[Record]  
Convert the transmission to a list of records.

**assignments**  
List of assignments.

**data\_recipient**  
Data recipient's Nets ID. String of 8 digits.

**data\_transmitter**  
Data transmitter's Nets ID. String of 8 digits.

**date**  
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**  
Data transmitters unique enumeration of the transmission. String of 7 digits.

## 2.2.2 Assignment class

**class** netsgiro.Assignment (*service\_code: Union[netsgiro.enums.ServiceCode, int, str], type: Union[netsgiro.enums.AssignmentType, int, str], number: str, account: str, agreement\_id: Optional[str] = None, date: Optional[datetime.date] = None, transactions: List[Union[Transaction, Agreement, PaymentRequest]] = 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.PaymentRequest

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, validate\_due\_date: bool = False*) → netsgiro.objects.PaymentRequest

Add an AvtaleGiro payment request to the assignment.

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

**classmethod from\_records** (*records: List[R]*) → 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[Union[netsgiro.records.AssignmentStart, netsgiro.records.TransactionAmountItem1, netsgiro.records.TransactionAmountItem2, netsgiro.records.TransactionAmountItem3, netsgiro.records.AssignmentEnd]]  
 Convert the assignment to a list of records.

**account**  
 The payee's bank account. String of 11 digits.

**agreement\_id**  
 Used for OCR Giro.  
 The payee's agreement ID with Nets. String of 9 digits.

**date**  
 Used for OCR Giro.  
 The date the assignment was generated by Nets.

**number**  
 The assignment number. String of 7 digits.

**service\_code**  
 The service code. One of *ServiceCode*.

**transactions**  
 List of transaction objects, like *Agreement*, *PaymentRequest*, *Transaction*.

**type**  
 The transaction type. One of *TransactionType*.

### 2.2.3 Transaction classes

**class** netsgiro.**Agreement** (*service\_code: Union[netsgiro.enums.ServiceCode, int, str]*, *number: int*, *registration\_type: Union[netsgiro.enums.AvtaleGiroRegistrationType, int, str]*, *kid: Optional[str]*, *notify: bool*)

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.AvtaleGiroAgreement]*) → netsgiro.objects.Agreement  
 Build an Agreement object from a list of record objects.

**to\_records** () → Iterable[netsgiro.records.AvtaleGiroAgreement]  
 Convert the agreement to a list of records.

**TRANSACTION\_TYPE = 94**

**kid**

KID number to identify the customer and invoice.

**notify**

Whether the payer wants notification about payment requests.

**number**

Transaction number. Unique and ordered within an assignment.

**registration\_type**

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

**service\_code**

The service code. One of *ServiceCode*.

```
class netsgiro.PaymentRequest (service_code: Union[netsgiro.enums.ServiceCode, int, str],
                               type: Union[netsgiro.enums.TransactionType, int, str], number: int,
                               date: datetime.date, amount, kid: Optional[str], reference: Optional[str],
                               text: Optional[str], payer_name: Optional[str])
```

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[TR]) → netsgiro.objects.PaymentRequest
```

Build a Transaction object from a list of record objects.

```
to_records () → Iterable[Union[netsgiro.records.TransactionAmountItem1, netsgiro.records.TransactionAmountItem2,
                                netsgiro.records.TransactionSpecification]]
```

Convert the transaction to a list of records.

**amount**

Transaction amount in NOK with two decimals.

**amount\_in\_cents**

Transaction amount in NOK cents.

**date**

The due date.

**kid**

KID number to identify the customer and invoice.

**number**

Transaction number. Unique and ordered within an assignment.

**payer\_name**

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**

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

**service\_code**

The service code. One of *ServiceCode*.

**text**

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**

The transaction type. One of *TransactionType*.

```
class netsgiro.Transaction (service_code: Union[netsgiro.enums.ServiceCode, int, str],
                             type: Union[netsgiro.enums.TransactionType, int, str], number: int,
                             date: datetime.date, amount, kid: Optional[str],
                             reference: Optional[str], text: Optional[str], centre_id: Optional[str],
                             day_code: Optional[int], partial_settlement_number: Optional[int],
                             partial_settlement_serial_number: Optional[str],
                             sign: Optional[str], form_number: Optional[str], bank_date: Optional[datetime.date],
                             debit_account: Optional[str], filler: Optional[str])
```

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[TR]) → netsgiro.objects.Transaction
    Build a Transaction object from a list of record objects.
```

```
to_records () → Iterable[Union[netsgiro.records.TransactionAmountItem1, nets-
                               giro.records.TransactionAmountItem2, netsgiro.records.TransactionAmountItem3]]
    Convert the transaction to a list of records.
```

**amount**  
Transaction amount in NOK with two decimals.

**amount\_in\_cents**  
Transaction amount in NOK cents.

**bank\_date**  
Used for OCR Giro.

**centre\_id**  
Used for OCR Giro.

**date**  
Nets' processing date.

**day\_code**  
Used for OCR Giro.

**debit\_account**  
Used for OCR Giro.

**form\_number**  
Used for OCR Giro.

**kid**  
KID number to identify the customer and invoice.

**number**  
Transaction number. Unique and ordered within an assignment.

**partial\_settlement\_number**  
Used for OCR Giro.

**partial\_settlement\_serial\_number**  
Used for OCR Giro.

**reference**  
The value depends on the payment method.

**service\_code**  
The service code. One of *ServiceCode*.

**sign**  
Used for OCR Giro.

**text**  
Up to 40 chars of free text from the payment terminal.



(continued from previous page)

```
... NY2102310000006NAVN
↳00000
...↳
↳NY210088000000060000002000000000000006001706041706040000000000000000000000000
...↳
↳NY000089000000060000002200000000000006001706040000000000000000000000000000000
... ''.strip() # noqa
```

`netsgiro.records.parse()` parses the input and returns a record object for each line of input:

```
>>> import netsgiro.records
>>> records = netsgiro.records.parse(data)
>>> len(records)
22
>>> pprint(records)
[TransmissionStart(service_code=<ServiceCode.NONE: 0>, transmission_number='1000081
↳', data_transmitter='55555555', data_recipient='00008080'),
AssignmentStart(service_code=<ServiceCode.AVTALEGIRO: 21>, assignment_type=
↳<AssignmentType.TRANSACTIONS: 0>, assignment_number='4000086', assignment_
↳account='8888888888', agreement_id='000000000'),
TransactionAmountItem1(service_code=<ServiceCode.AVTALEGIRO: 21>, transaction_
↳type=<TransactionType.PURCHASE_WITH_TEXT: 21>, transaction_number=1, nets_
↳date=datetime.date(2004, 6, 17), amount=100, kid='008000011688373', centre_
↳id=None, day_code=None, partial_settlement_number=None, partial_settlement_
↳serial_number=None, sign=None),
TransactionAmountItem2(service_code=<ServiceCode.AVTALEGIRO: 21>, transaction_
↳type=<TransactionType.PURCHASE_WITH_TEXT: 21>, transaction_number=1,
↳reference=None, form_number=None, bank_date=None, debit_account=None, _
↳filler=None, payer_name='NAVN'),
TransactionSpecification(service_code=<ServiceCode.AVTALEGIRO: 21>, transaction_
↳type=<TransactionType.PURCHASE_WITH_TEXT: 21>, transaction_number=1, line_
↳number=1, column_number=1, text=' Gjelder Faktura: 168837 Dato: 19/03/04'),
TransactionSpecification(service_code=<ServiceCode.AVTALEGIRO: 21>, transaction_
↳type=<TransactionType.PURCHASE_WITH_TEXT: 21>, transaction_number=1, line_
↳number=1, column_number=2, text='                               ForfallsDato: 17/06/04'),
...
AssignmentEnd(service_code=<ServiceCode.AVTALEGIRO: 21>, assignment_type=
↳<AssignmentType.TRANSACTIONS: 0>, num_transactions=6, num_records=20, total_
↳amount=600, nets_date_1=datetime.date(2004, 6, 17), nets_date_2=datetime.
↳date(2004, 6, 17), nets_date_3=None),
TransmissionEnd(service_code=<ServiceCode.NONE: 0>, num_transactions=6, num_
↳records=22, total_amount=600, nets_date=datetime.date(2004, 6, 17))]
```

`netsgiro.records.parse(data: str) → List[R]`  
Parse an OCR file into a list of record objects.

## 2.3.2 Record types

Given a record object, all record fields are available as sensible Python types:

```
>>> assignment_end = records[-2]
>>> assignment_end.service_code
<ServiceCode.AVTALEGIRO: 21>
>>> assignment_end.RECORD_TYPE
<RecordType.ASSIGNMENT_END: 88>
>>> assignment_end.assignment_type
<AssignmentType.TRANSACTIONS: 0>
>>> assignment_end.nets_date_earliest
datetime.date(2004, 6, 17)
>>> assignment_end.nets_date_latest
```

(continues on next page)



```
class netsgiro.records.AssignmentStart (service_code: Union[netsgiro.enums.ServiceCode,  

                                         int, str], assignment_type:  

                                         Union[netsgiro.enums.AssignmentType, int, str],  

                                         assignment_number: str, assignment_account:  

                                         str, agreement_id: Optional[str] = None)
```

AssignmentStart is the first record of an assignment.

Each assignment can contain any number of transactions.

```
classmethod from_string (line: str) → R  

    Parse OCR string into a record object.
```

```
to_ocr () → str  

    Get record as OCR string.
```

```
RECORD_TYPE = 20
```

```
agreement_id
```

```
assignment_account
```

```
assignment_number
```

```
assignment_type
```

```
service_code
```

```
class netsgiro.records.AssignmentEnd (service_code: Union[netsgiro.enums.ServiceCode,  

                                         int, str], assignment_type:  

                                         Union[netsgiro.enums.AssignmentType, int, str],  

                                         num_transactions, num_records, total_amount:  

                                         Union[None, int, str] = None, nets_date_1:  

                                         Union[datetime.date, str, None] = None,  

                                         nets_date_2: Union[datetime.date, str, None]  

                                         = None, nets_date_3: Union[datetime.date, str,  

                                         None] = None)
```

AssignmentEnd is the last record of an assignment.

```
classmethod from_string (line: str) → R  

    Parse OCR string into a record object.
```

```
to_ocr () → str  

    Get record as OCR string.
```

```
RECORD_TYPE = 88
```

```
assignment_type
```

```
nets_date
```

Nets' processing date.

Only used for OCR Giro.

```
nets_date_1
```

```
nets_date_2
```

```
nets_date_3
```

```
nets_date_earliest
```

Earliest date from the contained transactions.

```
nets_date_latest
```

Latest date from the contained transactions.

```
num_records
```

```
num_transactions
```

```
service_code
```



`total_amount`

```
class netsgiro.records.TransactionAmountItem1 (service_code:
    Union[netsgiro.enums.ServiceCode,
    int, str], transaction_type:
    Union[netsgiro.enums.TransactionType,
    int, str], transaction_number,
    nets_date: Union[datetime.date,
    str], amount, kid: Optional[str],
    centre_id: Optional[str] = None,
    day_code: Union[None, int, str] =
    None, partial_settlement_number:
    Union[None, int, str] = None,
    partial_settlement_serial_number:
    Optional[str] = None, sign: Op-
    tional[str] = 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) → R
    Parse OCR string into a record object.
```

```
to_ocr () → str
    Get record as OCR string.
```

```
RECORD_TYPE = 30
```

`amount`

`centre_id`

`day_code`

`kid`

`nets_date`

`partial_settlement_number`

`partial_settlement_serial_number`

`service_code`

`sign`

`transaction_number`

`transaction_type`

```
class netsgiro.records.TransactionAmountItem2 (service_code:
    Union[netsgiro.enums.ServiceCode,
    int, str], transaction_type:
    Union[netsgiro.enums.TransactionType,
    int, str], transaction_number, refer-
    ence: Optional[str], form_number:
    Optional[str] = None, bank_date:
    Union[datetime.date, str, None] =
    None, debit_account: Optional[str]
    = None, filler: Optional[str] = None,
    payer_name: Optional[str] = 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) → R
    Parse OCR string into a record object.
```

`to_ocr () → str`  
 Get record as OCR string.

`RECORD_TYPE = 31`

`bank_date`

`debit_account`

`form_number`

`payer_name`

`reference`

`service_code`

`transaction_number`

`transaction_type`

```
class netsgiro.records.TransactionAmountItem3 (service_code:
                                             Union[netsgiro.enums.ServiceCode,
                                             int, str], transaction_type:
                                             Union[netsgiro.enums.TransactionType,
                                             int, str], transaction_number, text:
                                             Optional[str])
```

TransactionAmountItem3 is the third record of a transaction.

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

`classmethod from_string (line: str) → R`  
 Parse OCR string into a record object.

`to_ocr () → str`  
 Get record as OCR string.

`RECORD_TYPE = 32`

`service_code`

`text`

`transaction_number`

`transaction_type`

```
class netsgiro.records.TransactionSpecification (service_code:
                                                Union[netsgiro.enums.ServiceCode,
                                                int, str], transaction_type:
                                                Union[netsgiro.enums.TransactionType,
                                                int, str], 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) → R`  
 Parse OCR string into a record object.

`classmethod from_text (*, service_code: netsgiro.enums.ServiceCode, transaction_type:
 TransactionType, transaction_number: int, text: Optional[str]) →
 Iterable[TransactionSpecification]`  
 Create a sequence of specification records from a text string.

`to_ocr () → str`  
 Get record as OCR string.

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

**RECORD\_TYPE** = 49

**column\_number**

**line\_number**

**service\_code**

**text**

**transaction\_number**

**transaction\_type**

```
class netsgiro.records.AvtaleGiroAgreement (service_code:
                                         Union[netsgiro.enums.ServiceCode,
                                         int, str], transaction_type:
                                         Union[netsgiro.enums.TransactionType,
                                         int, str], transaction_number,
                                         registration_type:
                                         Union[netsgiro.enums.AvtaleGiroRegistrationType,
                                         int, str], kid: Optional[str], notify:
                                         Union[bool, str])
```

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*) → R  
 Parse OCR string into a record object.

**to\_ocr** () → str  
 Get record as OCR string.

**RECORD\_TYPE** = 70

**kid**

**notify**

**registration\_type**

**service\_code**

**transaction\_number**

**transaction\_type**

## 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 v2.0.0 (2022-05-03)

#### Breaking

- Dropped support for Python 3.5 and 3.6 (both have reached EOL).

#### New

- Add support for Python 3.9, 3.10, and added 3.11 pre-releases to test-suite.
- Fully annotated the project and added a `py.typed` to make the package PEP 561-compliant.
- Added a `get_minimum_due_date` utility function for retrieving the earliest valid due date for a payment request.

- Added a `validate_due_date` kwarg to the `add_payment_request` Assignment method, to optionally perform due date validation. When enabled, due dates less than the minimum (4 calendar days in the future, offset by holidays) or more than the maximum (12 months in the future) will raise an exception.

### Internal

- Switched CI pipeline from CircleCI to Github actions.
- Added `isort`, `black`, `pyupgrade`, and `pre-commit` for linting.
- Switched to `Poetry` for package and dependency management.

### 2.5.2 v1.3.0 (2020-06-10)

- Drop support for Python 3.4, which has reached end of life.
- Python 3.8 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, 3.7 and 3.8.

### 2.5.3 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`.

### 2.5.4 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.

### 2.5.5 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.

### 2.5.6 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.

### 2.5.7 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.

### 2.5.8 v1.0.0a1 (2017-04-17)

Initial alpha release. No promises about backwards compatibility.





**n**

`netsgiro`, [12](#)

`netsgiro.records`, [17](#)



## A

- account (*netsgiro.Assignment* attribute), 14
- ACTIVE\_AGREEMENT (*netsgiro.AvtaleGiroRegistrationType* attribute), 25
- 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
- agreement\_id (*netsgiro.records.AssignmentStart* attribute), 20
- amount (*netsgiro.PaymentRequest* attribute), 15
- amount (*netsgiro.records.TransactionAmountItem1* attribute), 21
- amount (*netsgiro.Transaction* attribute), 16
- amount\_in\_cents (*netsgiro.PaymentRequest* attribute), 15
- amount\_in\_cents (*netsgiro.Transaction* attribute), 16
- Assignment (*class in netsgiro*), 13
- assignment\_account (*netsgiro.records.AssignmentStart* attribute), 20
- ASSIGNMENT\_END (*netsgiro.RecordType* attribute), 24
- assignment\_number (*netsgiro.records.AssignmentStart* attribute), 20
- ASSIGNMENT\_START (*netsgiro.RecordType* attribute), 24
- assignment\_type (*netsgiro.records.AssignmentEnd* attribute), 20
- assignment\_type (*netsgiro.records.AssignmentStart* attribute), 20
- AssignmentEnd (*class in netsgiro.records*), 20
- assignments (*netsgiro.Transmission* attribute), 13
- AssignmentStart (*class in netsgiro.records*), 19
- AssignmentType (*class in netsgiro*), 24
- AVTALEGIRO (*netsgiro.ServiceCode* attribute), 23
- AVTALEGIRO\_AGREEMENT (*netsgiro.TransactionType* attribute), 25
- AVTALEGIRO\_AGREEMENTS (*netsgiro.AssignmentType* attribute), 24
- AVTALEGIRO\_CANCELLATION (*netsgiro.TransactionType* attribute), 25
- AVTALEGIRO\_CANCELLATIONS (*netsgiro.AssignmentType* attribute), 24
- AVTALEGIRO\_WITH\_BANK\_NOTIFICATION (*netsgiro.TransactionType* attribute), 25
- AVTALEGIRO\_WITH\_PAYEE\_NOTIFICATION (*netsgiro.TransactionType* attribute), 25
- AvtaleGiroAgreement (*class in netsgiro.records*), 23
- AvtaleGiroRegistrationType (*class in netsgiro*), 25

## B

- bank\_date (*netsgiro.records.TransactionAmountItem2* attribute), 22
- bank\_date (*netsgiro.Transaction* attribute), 16

## C

- centre\_id (*netsgiro.records.TransactionAmountItem1* attribute), 21
- centre\_id (*netsgiro.Transaction* attribute), 16
- column\_number (*netsgiro.records.TransactionSpecification* attribute), 23

## D

- data\_recipient (*netsgiro.records.TransmissionStart* attribute), 19
- data\_recipient (*netsgiro.Transmission* attribute), 13
- data\_transmitter (*netsgiro.records.TransmissionStart* attribute), 19
- 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.records.TransactionAmountItem1* attribute), 21  
 day\_code (*netsgiro.Transaction* attribute), 16  
 debit\_account (*netsgiro.records.TransactionAmountItem2* attribute), 22  
 debit\_account (*netsgiro.Transaction* attribute), 16  
 DELETED\_AGREEMENT (*netsgiro.AvtaleGiroRegistrationType* attribute), 25

## F

form\_number (*netsgiro.records.TransactionAmountItem2* attribute), 22  
 form\_number (*netsgiro.Transaction* attribute), 16  
 FROM\_AVTALEGIRO (*netsgiro.TransactionType* attribute), 24  
 FROM\_BUSINESS\_TERMINAL\_GIRO (*netsgiro.TransactionType* attribute), 24  
 FROM\_CASH\_GIRO (*netsgiro.TransactionType* attribute), 24  
 FROM\_COUNTER\_GIRO (*netsgiro.TransactionType* attribute), 24  
 FROM\_DIRECT\_REMITTANCE (*netsgiro.TransactionType* attribute), 24  
 FROM\_GIRO\_DEBITED\_ACCOUNT (*netsgiro.TransactionType* attribute), 24  
 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), 16  
 from\_records() (*netsgiro.Transmission* class method), 12  
 FROM\_STANDING\_ORDERS (*netsgiro.TransactionType* attribute), 24  
 from\_string() (*netsgiro.records.AssignmentEnd* class method), 20  
 from\_string() (*netsgiro.records.AssignmentStart* class method), 20  
 from\_string() (*netsgiro.records.AvtaleGiroAgreement* class method), 23  
 from\_string() (*netsgiro.records.TransactionAmountItem1* class method), 21  
 from\_string() (*netsgiro.records.TransactionAmountItem2* class method), 21  
 from\_string() (*netsgiro.records.TransactionAmountItem3*

class method), 22  
 from\_string() (*netsgiro.records.TransactionSpecification* class method), 22  
 from\_string() (*netsgiro.records.TransmissionEnd* class method), 19  
 from\_string() (*netsgiro.records.TransmissionStart* class method), 19  
 FROM\_TELEGIRO (*netsgiro.TransactionType* attribute), 24  
 from\_text() (*netsgiro.records.TransactionSpecification* class method), 22

## G

get\_earliest\_transaction\_date() (*netsgiro.Assignment* method), 14  
 get\_latest\_transaction\_date() (*netsgiro.Assignment* method), 14  
 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), 15  
 kid (*netsgiro.PaymentRequest* attribute), 15  
 kid (*netsgiro.records.AvtaleGiroAgreement* attribute), 23  
 kid (*netsgiro.records.TransactionAmountItem1* attribute), 21  
 kid (*netsgiro.Transaction* attribute), 16

## L

line\_number (*netsgiro.records.TransactionSpecification* attribute), 23

## N

nets\_date (*netsgiro.records.AssignmentEnd* attribute), 20  
 nets\_date (*netsgiro.records.TransactionAmountItem1* attribute), 21  
 nets\_date (*netsgiro.records.TransmissionEnd* attribute), 19  
 nets\_date\_1 (*netsgiro.records.AssignmentEnd* attribute), 20  
 nets\_date\_2 (*netsgiro.records.AssignmentEnd* attribute), 20

nets\_date\_3 (*netsgiro.records.AssignmentEnd* attribute), 20  
 nets\_date\_earliest (*netsgiro.records.AssignmentEnd* attribute), 20  
 nets\_date\_latest (*netsgiro.records.AssignmentEnd* attribute), 20  
 netsgiro (*module*), 12  
 netsgiro.NETS\_ID (*built-in variable*), 23  
 netsgiro.records (*module*), 17  
 NEW\_OR\_UPDATED\_AGREEMENT (*netsgiro.AvtaleGiroRegistrationType* attribute), 25  
 NONE (*netsgiro.ServiceCode* attribute), 23  
 notify (*netsgiro.Agreement* attribute), 15  
 notify (*netsgiro.records.AvtaleGiroAgreement* attribute), 23  
 num\_records (*netsgiro.records.AssignmentEnd* attribute), 20  
 num\_records (*netsgiro.records.TransmissionEnd* attribute), 19  
 num\_transactions (*netsgiro.records.AssignmentEnd* attribute), 20  
 num\_transactions (*netsgiro.records.TransmissionEnd* attribute), 19  
 number (*netsgiro.Agreement* attribute), 15  
 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), 23

## P

parse() (*in module netsgiro*), 12  
 parse() (*in module netsgiro.records*), 18  
 partial\_settlement\_number (*netsgiro.records.TransactionAmountItem1* attribute), 21  
 partial\_settlement\_number (*netsgiro.Transaction* attribute), 16  
 partial\_settlement\_serial\_number (*netsgiro.records.TransactionAmountItem1* attribute), 21  
 partial\_settlement\_serial\_number (*netsgiro.Transaction* attribute), 16  
 payer\_name (*netsgiro.PaymentRequest* attribute), 15  
 payer\_name (*netsgiro.records.TransactionAmountItem2* attribute), 22  
 PaymentRequest (*class in netsgiro*), 15  
 PURCHASE\_WITH\_KID (*netsgiro.TransactionType* attribute), 25

PURCHASE\_WITH\_TEXT (*netsgiro.TransactionType* attribute), 25

## R

RECORD\_TYPE (*netsgiro.records.AssignmentEnd* attribute), 20  
 RECORD\_TYPE (*netsgiro.records.AssignmentStart* attribute), 20  
 RECORD\_TYPE (*netsgiro.records.AvtaleGiroAgreement* attribute), 23  
 RECORD\_TYPE (*netsgiro.records.TransactionAmountItem1* attribute), 21  
 RECORD\_TYPE (*netsgiro.records.TransactionAmountItem2* attribute), 22  
 RECORD\_TYPE (*netsgiro.records.TransactionAmountItem3* attribute), 22  
 RECORD\_TYPE (*netsgiro.records.TransactionSpecification* attribute), 23  
 RECORD\_TYPE (*netsgiro.records.TransmissionEnd* attribute), 19  
 RECORD\_TYPE (*netsgiro.records.TransmissionStart* attribute), 19  
 RecordType (*class in netsgiro*), 24  
 reference (*netsgiro.PaymentRequest* attribute), 15  
 reference (*netsgiro.records.TransactionAmountItem2* attribute), 22  
 reference (*netsgiro.Transaction* attribute), 16  
 registration\_type (*netsgiro.Agreement* attribute), 15  
 registration\_type (*netsgiro.records.AvtaleGiroAgreement* attribute), 23  
 REVERSING\_WITH\_KID (*netsgiro.TransactionType* attribute), 25  
 REVERSING\_WITH\_TEXT (*netsgiro.TransactionType* attribute), 25

## S

service\_code (*netsgiro.Agreement* attribute), 15  
 service\_code (*netsgiro.Assignment* attribute), 14  
 service\_code (*netsgiro.PaymentRequest* attribute), 15  
 service\_code (*netsgiro.records.AssignmentEnd* attribute), 20  
 service\_code (*netsgiro.records.AssignmentStart* attribute), 20  
 service\_code (*netsgiro.records.AvtaleGiroAgreement* attribute), 23  
 service\_code (*netsgiro.records.TransactionAmountItem1* attribute), 21

|  |  |
|--|--|
| service_code (netsgiro.records.TransactionAmountItem2 attribute), 22   | TRANSACTION_AGREEMENTS (netsgiro.RecordType attribute), 24                   |
| service_code (netsgiro.records.TransactionAmountItem3 attribute), 22   | TRANSACTION_AMOUNT_ITEM_1 (netsgiro.RecordType attribute), 24                |
| service_code (netsgiro.records.TransactionSpecification attribute), 23 | TRANSACTION_AMOUNT_ITEM_2 (netsgiro.RecordType attribute), 24                |
| service_code (netsgiro.records.TransmissionEnd attribute), 19          | TRANSACTION_AMOUNT_ITEM_3 (netsgiro.RecordType attribute), 24                |
| service_code (netsgiro.records.TransmissionStart attribute), 19        | transaction_number (netsgiro.records.AvtaleGiroAgreement attribute), 23      |
| service_code (netsgiro.Transaction attribute), 16                      | transaction_number (netsgiro.records.TransactionAmountItem1 attribute), 21   |
| ServiceCode (class in netsgiro), 23                                    | transaction_number (netsgiro.records.TransactionAmountItem2 attribute), 22   |
| sign (netsgiro.records.TransactionAmountItem1 attribute), 21           | transaction_number (netsgiro.records.TransactionAmountItem3 attribute), 22   |
| sign (netsgiro.Transaction attribute), 16                              | transaction_number (netsgiro.records.TransactionSpecification attribute), 23 |
| <b>T</b>   | TRANSACTION_SPECIFICATION (netsgiro.RecordType attribute), 24                |
| text (netsgiro.PaymentRequest attribute), 15                           | TRANSACTION_TYPE (netsgiro.Agreement attribute), 14                          |
| text (netsgiro.records.TransactionAmountItem3 attribute), 22           | transaction_type (netsgiro.records.AvtaleGiroAgreement attribute), 23        |
| text (netsgiro.records.TransactionSpecification attribute), 23         | transaction_type (netsgiro.records.TransactionAmountItem1 attribute), 21     |
| text (netsgiro.Transaction attribute), 16                              | transaction_type (netsgiro.records.TransactionAmountItem2 attribute), 22     |
| to_ocr() (netsgiro.records.AssignmentEnd method), 20                   | transaction_type (netsgiro.records.TransactionAmountItem3 attribute), 22     |
| to_ocr() (netsgiro.records.AssignmentStart method), 20                 | transaction_type (netsgiro.records.TransactionSpecification attribute), 23   |
| to_ocr() (netsgiro.records.AvtaleGiroAgreement method), 23             | TransactionAmountItem1 (class in netsgiro.records), 21                       |
| to_ocr() (netsgiro.records.TransactionAmountItem1 method), 21          | TransactionAmountItem2 (class in netsgiro.records), 21                       |
| to_ocr() (netsgiro.records.TransactionAmountItem2 method), 21          | TransactionAmountItem3 (class in netsgiro.records), 22                       |
| to_ocr() (netsgiro.records.TransactionAmountItem3 method), 22          | transactions (netsgiro.Assignment attribute), 14                             |
| to_ocr() (netsgiro.records.TransactionSpecification method), 22        | TRANSACTIONS (netsgiro.AssignmentType attribute), 24                         |
| to_ocr() (netsgiro.records.TransmissionEnd method), 19                 | TransactionSpecification (class in netsgiro.records), 22                     |
| to_ocr() (netsgiro.records.TransmissionStart method), 19               | TransactionType (class in netsgiro), 24                                      |
| to_ocr() (netsgiro.Transaction method), 13                             | Transmission (class in netsgiro), 12   |
| to_records() (netsgiro.Agreement method), 14                           | TRANSMISSION_END (netsgiro.RecordType attribute), 24                         |
| to_records() (netsgiro.Assignment method), 14                          | transmission_number (netsgiro.TransactionSpecification attribute), 23        |
| to_records() (netsgiro.PaymentRequest method), 15                      | TransactionAmountItem1 (class in netsgiro.records), 21                       |
| to_records() (netsgiro.Transaction method), 16                         | TransactionAmountItem2 (class in netsgiro.records), 21                       |
| to_records() (netsgiro.Transmission method), 13                        | TransactionAmountItem3 (class in netsgiro.records), 22                       |
| to_text() (netsgiro.records.TransactionSpecification class method), 23 | TRANSACTIONS (netsgiro.AssignmentType attribute), 24                         |
| total_amount (netsgiro.records.AssignmentEnd attribute), 20            | TransactionSpecification (class in netsgiro.records), 22                     |
| total_amount (netsgiro.records.TransmissionEnd attribute), 19          | TransactionType (class in netsgiro), 24                                      |
| Transaction (class in netsgiro), 15                                    | Transmission (class in netsgiro), 12   |
|  | TRANSMISSION_END (netsgiro.RecordType attribute), 24                         |
|  | transmission_number (netsgiro.TransactionSpecification attribute), 23        |

*giro.records.TransmissionStart* attribute),  
19  
TRANSMISSION\_START (*netsgiro.RecordType* at-  
tribute), 24  
TransmissionEnd (*class in netsgiro.records*), 19  
TransmissionStart (*class in netsgiro.records*), 19  
type (*netsgiro.Assignment* attribute), 14  
type (*netsgiro.PaymentRequest* attribute), 15  
type (*netsgiro.Transaction* attribute), 16