

---

# **CWR-API Documentation**

***Release 0.0.8***

**Bernardo Martinez Garrido**

**Sep 30, 2017**



---

## Contents

---

<b>1</b>	<b>Getting the library</b>	<b>3</b>
<b>2</b>	<b>Contents</b>	<b>5</b>
2.1	Documentation for CWR-DataApi . . . . .	5
2.2	CWR Data API . . . . .	18
2.3	Indices and tables . . . . .	21



**Author** Bernardo Martínez Garrido

**Copyright** WESO 2015

**License** [MIT](#)

**Interpreters** Python 2.6, 2.7, 3.2, 3.3, 3.4, Pypy, Pypy3

The CWR API library offers a model to represent the content of files following the CISAC CWR standard.

With this model, and various helper classes, it is not only possible to read and show those files, but also to operate with the data they contain.

This library has been developed based on CWR specification version 2.1 revision 3, from December 10th 2004.



# CHAPTER 1

---

## Getting the library

---

The library can be found at Pypi, making its installation it very easy:

```
$ pip install cwr-api
```



## Documentation for CWR-DataApi

Contents:

### CWR Standard

#### The CWR standard

The CWR file standard has been created by CISAC as a way for publishers and societies to share discographic information.

CWR files containing updated information are created, sent by a sending party to a receiver, and then processed before an acknowledgement file is returned.

#### Documents

While the CWR standard is defined on a single document CISAC has created a few other which may help to understand it or, in the case of for example the lookup tables, are actually required if one wishes to work with the files.

The files which can be downloaded here are the ones which have been used to develop the library. But the latest version of these files (which at the time of creating the project are the same on this page) can always be found on the {{{www.cisac.org}CISAC}} website.

#### Specification

A single file contains the standard specification:

- Functional specifications: Common Works Registration version 2.1. - Rev.7 (CWR11-1991R1)

Additional information about rules to be used when writing into a file can be found on the following files:

- CIS Character Set Rules (CWR11-1494)
- Amendments of TIS data (TIS09-1546R1)

### Manuals

There is a user manual to help when working with these files:

- CWR v2.1 User Manual (CWR11-1494)

### Data Files

Some files contain information required to fill and validate the CWR files.

- CWR Validation and Lookup Tables (CRF020)
- CWR Sender ID and Codes (CWR06-1972)
- CWR Error Messages (CWR08-2493)
- EDI Standards (IM0047) (MISSING)

TIS information.

- TIS Territories (TIS09-1540a)
- TIS Hierarchies (TIS09-1540c)

### Other Files

Miscellany related files. Some of these are not required to use the standard, but can help to understand it, while others have a very specific use.

The CWR light is a variant which reduces greatly the number of fields in the records.

- CWR Light Specifications (CWR08-3540)

The following files are used to prepare a CWR-based communication between two parties.

- CWR Publisher Questionnaire (CWR07-1411)
- CWR Society Questionnaire (CWR08-1983)

There is an implementation spreadsheet showing society and publisher communications have been established with the CWR standard.

- CWR Implementation Spreadsheet (CWR06-1950)

### CWR file structure

CWR files are meant to be used as an information transmission system and reflect this on their structure.

The information is divided in two parts: the file name contains data uniquely identifying the file, and the contents of the file contain the actual information being sent.

## File name

The CWR standard gives a special importance to the file's name, certain metadata is stored on it to allow it being used as an unique identifier for the file.

The filename follows the pattern CWyynnnnsss\_rrr.Vxx where each section means the following:

- CW: Header indicating it is a CWR file.
- yy: Year.
- nnnn: Sequence.
- sss: Sender. 2 or 3 digits.
- rrr: Receiver. 2 or 3 digits.
- xx: Version of the CWR standard (version x.x).

On the original CWR v2.1 specification the sequence number consists of only two digits. Was changed to four on revision 5.

Note that the filename specification is not always followed, and so by default it should be considered optional.

## File structure

The file is structured as a batch process, storing a consecutive series of transactions.

All these transactions are grouped into a single Transmission, which is then divided into several Groups, one for each type of transaction.

The start and ending of both the Transmission and the Groups are marked by a header and trailer record.

So the structure of file's interior can be defined as: [HDR, [GRH, GRT]\*, TRL]

Where the tags are the CWR record header tags, meaning:

- HDR: Transmission header
- TRL: Transmission trailer
- GRH: Group header
- GRT: Group trailer

These four types of record are the Control Records of the file, used not only to separate the sections, but also to verify the data contained in them is correct.

This is done comparing the information these records contain with the information read from the section they enclose.

## Transmission

There is only single Transmission in the file, and it contains all the records.

## Groups

Groups contain batches of transactions.

While there are several groups on the file, there can be only one for each type of transaction, and they indicate which type of transaction they are storing.

They are numbered consecutively, starting on 1. No two Groups may have the same number, and there can't be any gaps between their numbers.

### Transaction

A Transaction is a batch of records containing all the data for a single job.

For example, a Transaction may contain information for registering an Agreement, for indicating a registering conflict with a Work, or even for indicating an error on a Transaction.

A Transaction is always of a single type, which is specified by its header record, and will be named after it. So if a Transaction starts with an Agreement record it is an Agreement Transaction.

The possible transactions in CWR v2.1 are:

- Acknowledgment of Transaction (ACK)
- Agreement supporting Work Registration (AGR)
- Existing Work which is in conflict with a Work registration (EXC)
- New Works Registration (NWR)
- Notification of ISWC assigned to a Work (ISW)
- Revised Registration (REV)

In practise, a Transaction is just a relationship of Records, and it indicates which records can or should follow the header.

Going back to the previous example, an Agreement Transaction would indicate an Agreement, the Territories it applies to and the Interested Parties for each Territory.

### CWR field format

CWR records are divided into fields, each following a clear pattern, which consists on defining the following properties:

- Field name
- Start index
- Size
- Format

The field name is used only to help humans identifying the field, while the start index and size serve to acquire the data from the record line.

## Formats

Format	Code	Notes
Alphanumeric	A	ASCII characters in upper case.
Boolean	B	A single character. Can be 'Y', for yes, or 'N', for no. Meaning the Boolean values True and False.
Flag	F	The same as Boolean, but adding a third option: 'U' for unknown. This can't be parsed into a Boolean value.
Date	D	Eight numeric characters, following the pattern YYYYMMDD.
Numeric	N	Usually an integer, sometimes a float value. In that case the field documentation indicates how many characters are for the decimal value.
Time	T	Six numeric characters, following the pattern HHMMSS. It is on military format (24 hours, and not two groups of 12).
List/Table Lookup	L	Only accepts values coming from a specific list or table. The table is indicated in the field description.

## Empty fields

Fields may be optional, or there may not be any data to put in them. In those cases, the empty columns must be filled as follows.

Format	Code	When empty
Alphanumeric	A	Columns should be filled with the empty character.
Boolean	B	?
Flag	F	Should be set as 'unknown' ('U').
Date	D	Columns should be set as 0.
Numeric	N	Columns should be set as 0.
Time	T	Columns should be set as 0.
List/Table Lookup	L	Columns should be filled with the empty character.

## Additional constraints

Date and Time formats have additional constraints due to the patterns they follow.

Date follows the pattern YYYYMMDD, which has the following constraints:

- YYYY: can be any number
- MM: ranges from 01 to 12
- DD: ranges from 01 to 31

Time follows the pattern HHMMSS, which has the following constraints:

- HH: ranges from 00 to 23
- MM: ranges from 00 to 59
- SS: ranges from 00 to 59

## CWR file validation

## File level validation

ID	Constraint	Failure level
1	File should be readable	ER
2	First record should be HDR	ER
3	Second record should be GRH	ER
4	Groups open with GRH and close with GRT	ER
5	Last record is TRL	ER
6	GRH should be followed by a Transaction header	ER
7	GRT should be followed by a GRH or TRL	ER
7	Only a single HDR and a single TRL exist	ER

## Acknowledgement file

After receiving and processing the CWR file, the recipient will create and return an acknowledgement file, containing most of the original file information, and adding Acknowledgement Transactions.

These transactions will include all additional information that may be needed, such as the reasons for rejecting a transaction, or the CAE/IPI numbers where they may be missing.

Information that is not relevant to the creator of the Acknowledgment file will not appear on it. For example, a society will generally not return SPU/SPT records for sub-publishers in territories it does not control.

Note that when validating the original CWR file the process won't stop at the first error encountered, but will continue to report all errors, unless a severe error makes further processing inadvisable.

## Acknowledgement report

According to the CWR standard, along the Acknowledgment file a form must be fulfilled and sent back to the submitter.

This contains the transmission participants:

- Society
- Sender (of the original CWR file)

And also a series of details:

- File name
- Location
- Description
- File size
- Date or time stamp (YYYYMMDD and HHMMSS format)
- Number of transactions and records

Along a series of boolean flags:

- The file has been received and is awaiting validation/processing
- The file has been received and has been successfully validated/processed
- The file is no longer required and can be deleted
- The file has been received and has failed validation/processing (It should be sent again and details of failure are to be indicated to the sender)

## Acknowledgement transaction

Information on the Acknowledgement file is added with the use of Acknowledgement transactions.

These mark the Transactions on the original file, adding any needed information about them, such as if it has been rejected.

It follows the structure: [ACK, MSG\*, AGR | NWR | REV | EXC]

## Other topics

### About the volumes of work

#### How many files may be sent and received, and how often?

As far as we know, the yearly volume is commonly quite low.

The current format allows sending up to ten thousand files between each submitter and recipient, which was increased from an original one hundred, but probably no submitter will send so many files.

CWR files are commonly sent by publishers each trimester, but can amount to up to four per year.

Societies on the other hand may not send a single CWR file in a year, but may reply to any and all received files with an acknowledgement file.

#### How many transactions may be contained in a file?

A single group may contain up to ten million transactions, and there can be a hundred thousand groups. But again, this limit seems hard to reach.

A big file will contain around a hundred thousand agreements, which will mean a few hundred thousand lines. While a smaller one will just have less than ten thousand lines.

## CWR Glossary

**Group** A collection with all the transactions of a type in the transmission.

**Control Records** Records used to ensure the data has not been damaged or tampered. These are the Transmission and Group Header and Trailer.

**File** In the CWR context, a file is one following the CWR standard, meaning it has the correct naming scheme and contents.

**Transaction** A collection of new information for things such as work registrations or agreements.

**Transaction Header** Initial record on a Transaction, which indicates the type of this transaction.

**Transmission** All the collected records in the file. It can be considered as the logical representation of all this data.

## Other standards and related information

The CWR standard makes use of other standards or sources of data, which are documented on this section.

## **IPI Number**

There are two fields which appear commonly in records referring to interested parties: IPI Name Number and IPI Base Number. These are references to ISAC's Interested Parties Information system, used by CISAC identify Interested Parties on collective rights management data, and are stored on their databases.

Sometimes this is referred as IPI/CAE number. But the CAE (Composer, Author and Publisher, the E standing for 'Editeur') system is obsolete since 2001.

These codes are used along ISWC (International Standard Musical Work Code) codes.

As previously indicated, IPI numbers are divided into two types:

- IPI Name Number
- IPI Base Number

## **IPI Name Number**

The IPI Name Number, is just composed of eleven numbers.

## **IPI Base Number**

The IPI Base Number, follows the pattern H-NNNNNNNNN-C, where each digit means:

- H: header, which is a single letter
- N: identification number. Nine numeric digits.
- C: check digit. A single number.

## **The CWR file structure in detail**

This section contains a more detailed view of how is data stored inside a CWR file.

## **Record Type Codes**

The valid Record types are always indicated in the latest CWR standard specification, and indicated on CISAC's Record Type table.

These are used on a Record prefix to identify its type.

This list is offered just to make it easier identifying each of them.

## **Control Records**

Record Type	Record name
GRH	Group Header
GRT	Group Trailer
HDR	Transmission Header
TRL	Transmission Trailer

## Transaction Records

Note that the Transaction type is actually the type of the header Record on the Transaction.

So for example an Acknowledgment Transaction starts with an Acknowledgment Record.

Record Type	Record name
ACK	Acknowledgment of Transaction
AGR	Agreement supporting Work Registration
EXC	Existing Work which is in conflict with a Work registration
NWR	New Works Registration
ISW	Notification of ISWC assigned to a Work
REV	Revised Registration

## Detail Records

Record Type	Record name
ALT	Alternate Title
ARI	Additional Related Information
COM	Composite Component
EWT	Entire Work Title for Excerpts
IPA	Interested Party of Agreement
IND	Instrumentation Detail
INS	Instrumentation Summary
MSG	Message
NAT	Non-Roman Alphabet Title
NCT	Non-Roman Alphabet Title for Components
NET	Non-Roman Alphabet Entire Work Title for Excerpts
NOW	Non-Roman Alphabet Other Writer Name
NPN	Non-Roman Alphabet Publisher Name
NPR	Performing Artist in Non-Roman alphabet
NVT	Non-Roman Alphabet Original Title for Versions
NWN	Non-Roman Alphabet Writer Name
OPU	Other Publisher
ORN	Work Origin
OWR	Other Writer
PER	Performing Artist
PWR	Publisher for Writer
REC	Recording Detail
SPT	Publisher Territory of Control
SPU	Publisher Controlled by Submitter
SWR	Writer Controlled by Submitter
SWT	Writer Territory of Control
TER	Territory in Agreement
VER	Original Work Title for Versions

## Record Prefix

All the records contain an initial field serving the uniquely identify them, and to note the type of record it is.

## Prefix according to the record

It should be noted that the prefix structure on this page applies only to Transaction and Detail records. Control records lack the Transaction Sequence Number and the Record Sequence Number.

## Structure of the prefix

This field is the record prefix, which contains just three values:

Field	Type	Description
Record Type	Table Lookup	One from the Record Type or Transaction Type tables
Transaction Sequence #	Numeric	Unique ID for each Transaction in a Group
Record Sequence #	Numeric	Unique ID for each Detail Record in a transaction

Transactions and Detail Records share both sequence numbers, but use them in a different way.

Transactions have the Record Sequence Number set to 0 always. Their Transaction Sequence Number is 0 for the first Transaction on a Group, and a consecutive value for the following Transactions on the same Group.

Detail Records use the Transaction Sequence Number of the Transaction they are part of, while the Record Sequence Number is that of the previous Record plus one.

Ambiguity: the specification file **is not** very clear about the Detail Records numbering. I suppose the first Detail Record on a Transaction should have RSN 1 (RSN of the Transaction, which **is** 0, plus 1)

## Example for sequence numbering

This would be an example of sequence numbering:

Record Type	Transaction Sequence #	Record Sequence #
First transaction header	0	0
First detail in transaction	0	1
Second detail in transaction	0	2
Third detail in transaction	0	3
Second transaction header	1	0
First detail in transaction	1	1
Second detail in transaction	1	2
Third transaction header	2	0
First detail in transaction	2	1
Second detail in transaction	2	2
Third detail in transaction	2	3
Fourth detail in transaction	2	4

## Transmission Header (HDR)

The Transmission header indicates the beginning of the CWR data on the file, and contains information about it's creation and sender.

It contains the following fields:

Field	Type	Required	Description
Record Type	Alphanumeric	Yes	It is always 'HDR'
Sender Type	Alphanumeric	Yes	Indicates the role of the sender. Only 'AA', 'PB', 'SO' or 'WR' are accepted.
Sender ID	Numeric	Yes	Code identifying the sender
Sender Name	Alphanumeric	Yes	Name of the sender
EDI Standard Version Number	Alphanumeric	Yes	Version of the header and trailer. '01.10' for CWR 2.1
Creation Date	Date	Yes	The date that this file was created
Creation Time	Time	Yes	The time of day that this file was created
Transmission Date	Time	Yes	The date that this file was transmitted to all receiving entities
Character Set	Time	No	To be used if this file contains data in a character set other than ASCII

### Transmission Trailer (TRL)

The Group Trailer closes a CWR file, and contains validation information.

This validation data is the number of groups, transactions and records which should have been processed.

It contains the following fields:

Field	Type	Required	Description
Record Type	Alphanumeric	Yes	It is always 'TRL'
Group Count	Numeric	Yes	Number of transactions in the Transmission
Transaction Count	Numeric	Yes	Number of transactions in the Transmission
Record Count	Numeric	Yes	Number of records in the Transmission

### Group Header (GRH)

The Group header indicates the beginning of a batch CWR transactions on the file.

It contains the following fields:

Field	Type	Required	Description
Record Type	Alphanumeric	Yes	It is always 'GRH'
Transaction Type	Table Lookup (Transaction Type table)	Yes	All the transactions in the group are of this type
Group ID	Numeric	Yes	Sequential ID starting on 1
Version Number	Alphanumeric	Yes	CWR version of the transaction. By default it is '02.10' for CWR 2.1
Batch request	Numeric	No	ID used by the submitter to internally identify this batch

### Group Trailer (GRT)

The Group Trailer closes a batch of transactions, and contains validation information.

This validation data is the number of transactions and records which should have been processed.

It contains the following fields:

Field	Type	Required	Description
Record Type	Alphanumeric	Yes	It is always 'GRT'
Group ID	Numeric	Yes	The same as the header
Transaction Count	Numeric	Yes	Number of transactions in the group
Record Count	Numeric	Yes	Number of records in the group

### **Acknowledgement Record (ACK)**

This record indicates the transaction status after its validation, along any information needed to link this transaction with the original one.

Field	Type	Re-quired	Description
Original Transaction Sequence #	Numeric	Yes	The sequence number of the original transaction
Original Transaction Type	Table Lookup (Transaction Type Table)	Yes	The type of the original transaction
Processing Date	Date	Yes	The date the file was received
Transaction Status	Table Lookup (Transaction Status Table)	Yes	Current status for the Transaction
Creation Title	Alphanumeric	No	If the original transaction refers to a work, its title should be here
Submitter Creation #	Numeric	No	ID assigned by the submitter. Required if the original Transaction was accepted
Recipient Creation #	Numeric	No	ID assigned by the recipient. Required if the original Transaction was accepted

### **Message Record (MSG)**

Indicates the results of validation and accompanies Acknowledgement records.

Field	Type	Re-quired	Description
Record Prefix	Alphanu- meric	Yes	The type is always 'MSG'
Message Type	List Lookup	Yes	One of 'F'/'R'/'T'/'G'/'E'
Original Record Sequence #	Numeric	Yes	The Record Sequence Number which caused this message
Record Type	Alphanu- meric	Yes	The Record Type which caused this message
Message Level	List Lookup	Yes	One of 'E'/'G'/'T'/'R'/'F'
Validation Number	Alphanu- meric	Yes	Identifies the specific edit condition that generated this message
Message Text	Alphanu- meric	Yes	The text associated with this message

### **CWR file validation**

This section contains a more detailed view of how is a CWR file validated.

## Validation process

Once received, the file undergoes a validation process.

Three levels of validation are applied:

- Transaction level
- Record level
- Field level

The Transaction validation ensures the overall relationship between the records is correct. This checks mainly the order, numbering and counts of records.

Record validation checks a concrete relationship between records. For example, a TER agreement should follow an AGR or TER agreement.

Field level validation ensures each field contains the correct information. This checks things such as the size of the field, the pattern it must follow, or that the references IDs exist somewhere.

## Failure levels

Each validation constraint has a failure level assigned.

Code	Failure name
ER	Entire file is rejected
GR	Entire group is rejected
TR	Entire transaction is rejected
RR	Entire record is rejected
FR	Record is rejected and set to the default value

## File validation

### File level validation

ID	Constraint	Failure level
1	File should be readable	ER
2	First record should be HDR	ER
3	Second record should be GRH	ER
4	Groups open with GRH and close with GRT	ER
5	Last record is TRL	ER
6	GRH should be followed by a Transaction header	ER
7	GRT should be followed by a GRH or TRL	ER
7	Only a single HDR and a single TRL exist	ER

### Record Prefix validation

## Field level validation

ID	Field	Constraint	Failure level
1	Record Type	Should be one from the Record Type or Transaction Type tables	ER
2,3,4,7,8	Transaction Sequence #	See below	TR/ER
5,6	Record Sequence #	See below	ER

## Transaction sequence numbering

- For the first Transaction header of a group it should be 0 (id 2, fl ER)
- For Transaction headers not being the first in the group, this is equal to the previous transaction number plus one (id 3, fl TR)
- For detail records the code is the same as the last transaction header (id 4, fl TR)
- Transactions sequence numbers should be sequential (id 7, fl ER)
- Detail records on a Transaction should have this Transaction's sequence number (id 8, fl ER)

## Record sequence numbering

- For Transaction headers it should be 0 (id 5, fl ER)
- For details records this is equal to the previous record number plus one (id 6, fl ER)

# CWR Data API

## CWR grammar

This section contains a more detailed view of how is the grammar for CWR files created.

## Rules

Depending on their scope, there are three kinds of rules:

- Terminal rules. These are fields, and are not composed by other rules.
- Record. These are the lines from the CWR files, and are composed of terminal rules.
- Group. These are aggregations of records. They may be composed of of any combination of rules, including other groups.

In practice only terminal rules are different. These are stored in Python modules, and so are static, while the other rules are generated dynamically from configuration files.

## Rules aggregation and trees

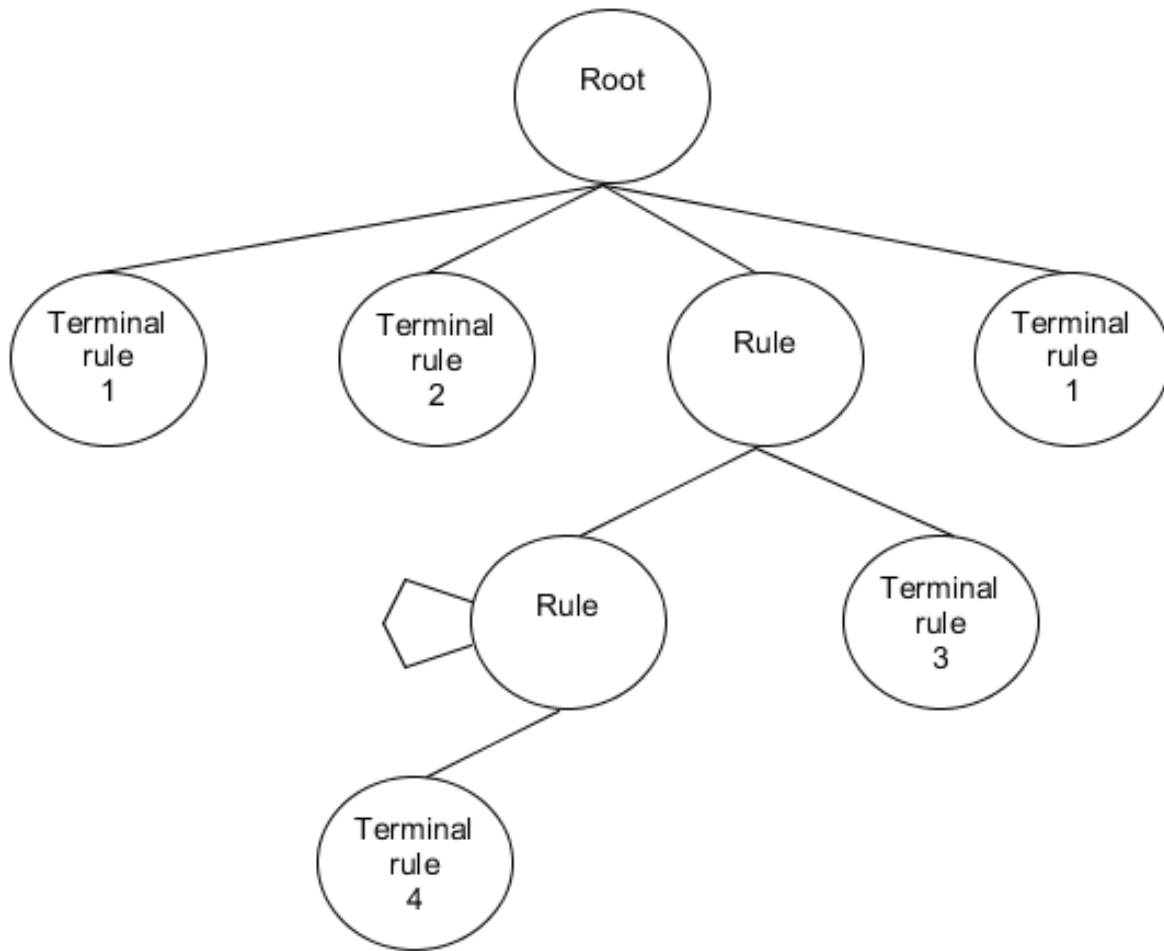
Except for terminal rules, all rules are an aggregation of smaller rules, creating a small tree.

These trees are read in pre-order.

## Examples

### Generic rules tree

The following example is a generic rules tree:



Rules nodes will be substituted by the rules they contain.

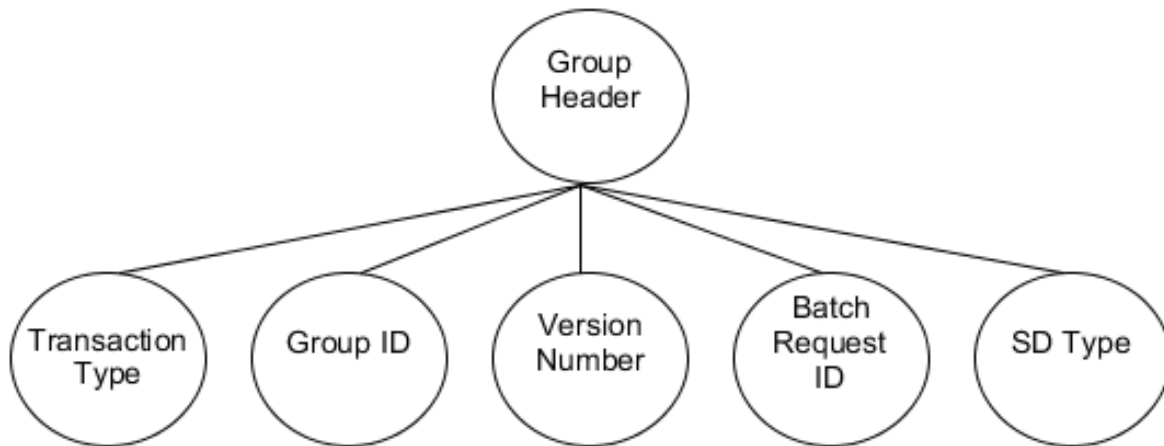
Note that the rule in the second level has a loop. This means that the rules it contains may appear multiple times.

As trees are read in pre-order, this would read as:

“Rule composed of terminal rule 1, followed by terminal rule 2, followed by terminal rule 4 multiple times, followed by terminal rule 3, followed by terminal rule 1.”

## Group Header

The following example represents the Group Header Record:



## Configuration DSL

A small DSL is being used to set up the grammar.

Files using this DSL are read and processed, and the data is then sent to the grammar factory to build the grammar.

An example of this DSL, defining the Agreement Record:

```
transaction_record:
  id: agreement
  head: AGR
  rules:
    [
      sequence
      [
        field: submitter_agreement_n
        field: international_standard_code
        field: agreement_type
        field: agreement_start_date
        field: agreement_end_date
        field: retention_end_date
        field: prior_royalty_status
        field: prior_royalty_start_date
        field: post_term_collection_status
        field: post_term_collection_end_date
        field: date_of_signature
        field: number_of_works
        field: sales_manufacture_clause
        field: shares_change
        field: advance_given
        field: society_assigned_agreement_n
      ]
    ]
]
```

## Rules composition

Rules are composed of several smaller rules. The terminal rules are the fields, defined on their own module.

This creates a tree of rules.

There are two groups of rules in a tree:

- Rules. Composed from a series of other rules.
- Rules lists. These are a set of rules, grouped by a combinatory rule.

Note that rules can be terminal rules. All rule blocks should generate trees ending in terminal rules.

## Rules tree

Show how the rules are defined as a tree.

## Structure

The DSL consists on a series of blocks, each of them representing a grammar rule.

These rules represent a logical section of the file, and may be for a line, or for a series of them.

They have the following structure, which only shows compulsory fields:

```
rule_group_1:
  id: rule_id_1
  rules:
    [
      internal_rules_list
      [
        rule_group_2: rule_id_2
        rule_group_1: rule_id_3
        rule_group_2: rule_id_4
      ]
      rule_group_2: rule_id_5
    ]
]
```

## Compulsory fields

Each block has a set of required fields:

Field	Notes
Root rule group	The root of the block. In the example it is 'transaction_record'. It indicates the global group to which it belongs.
Rule id	Identifier for this rule
Rules	The smaller rules which compose this rule
Internal rules	A new tree of rules
Rule group	A group of rules

## Indices and tables

- genindex

- [search](#)

## C

Control Records, [11](#)

## F

File, [11](#)

## G

Group, [11](#)

## T

Transaction, [11](#)

Transaction Header, [11](#)

Transmission, [11](#)