TRS – Technical Q&A

v1.3

Questions

Internal file reference

W:\TRS\NTRSII\Q&A\

Version control

Version	Date	Author	Comments
1.0	05/03/2019	Trond A. S. Andersen, Johan F. Øhman	First draft
1.1	11/03/2019	Trond A. S. Andersen	Added par. 4a)
1.2	20/03/2019	Trond A. S. Andersen	Added par. 2d)
1.3	26/03/2020	Trond A. S. Andersen	Added Chpt. 3

1. Report File Naming Convention(s)

a) How are TRS reporting files supposed to be named?

Additional to ESMA's general functional specification for transaction reporting according to MiFID art. 26, there are some specifics that solely apply to the Nordic countries and The Netherlands. These specifics mostly apply to validations on file level and the file naming convention scheme. As far as the file naming convention is concerned, the following requirements and principles constitutes the basis for the convention:

- Quickly and unambiguously distinguish between all reports files submitted from the industry.
- Filter out report files that are mal-formatted and hence unreliable early in the process, minimizing the need to perform content validation superfluously.

The naming convention for Transaction Report Files has been defined as follows:

TR_<SEIC>_<ORI>_<YYYYMMDD>_<RFSEQ>.<TYPE> (see breakdown of file name components in the table below)

Segment	Content
TR	Literal. Stands for "Transaction Report"
<SEIC>	Submitting Entity Identification Code. Legal entity identifier (LEI) as defined in ISO 17442 (20 alphanumerical characters). This is the same identifier that the transaction reports should comprise.
<ORI>	The **originating system or department** of the file. A two digit number.

Segment	Content
	00 = The TRSII system. Used for manual reporting via web form when the (TRSII) system creates the transaction report file. 01...99 = Department or system at the SE. Used for uploaded files or files sent from a SERS (automated). The number uniquely specify either the department that created the file manually, or the system that created and sent the file. *Rationale: One SE could have several systems and departments submitting reports. This allows the SE to keep the sequence number unique across systems and departments. This also makes investigations of reporting problems more efficient.*
<YYYYMMDD>	Date the file was created by the Submitting Entity. 8 digits in ISO 8601 format. Must be an existing date Must the same or an earlier date than the submission date.
<RFSEQ>	Report file sequence number. A 4 digit sequence number [0000-9999]. The system processes report files in batches, internally fetched from the (S)FTP area at certain intervals. Report files available at each interval constitutes one batch of files. Batches are processed in the order they are fetched from the FTP area. The report files in each batch are processed in sequence number order. Gaps in the sequence order within each batch are allowed and ignored. I.e., if report files 0004, 0005, 0008, 0009 are received and 0005 has been processed, the next report with a greater sequence number is processed: 0008. It is possible, and allowed, that a a report file in a batch has a lower sequence number than report files in previous batch. If a report file is rejected, processing continues with the next in sequence without interruption. This may cause subsequent errors with cancellations, which is an acceptable situation. *Rationale: It is more important to processing incoming reports than handling, for example, cancellations perfectly.* It is up to the SE to send the report files in the correct order and ensure that the correct transaction report version has been received by the FSA. Each day, numbering is allowed to restart at 0000, but that is not required.

Segment	Content
	Note on gaps: PreviousSequenceNumber has been considered but rejected. It is used for TREM since TREM is a middle-layer in between CAs. It is used to handle situation when TREM looses files, not when the sender (CA) misses something. In the local (industry) flow there is more a direct connection between the submitter (SE) and receiver (CA). Adding this restriction does not give a clear benefit. It adds functional complexity. The SE will know if the file was uploaded correctly and has the possibility to amend this to avoid non-deliberate gaps.
<TYPE>	File type. 3 to 5 characters. The file types that are accepted, is configurable. *Rationale: Countries allow different encryption methods and this should be possible to be reflected in the type.* The configuration is system-wide, i.e, applies to all SE in the country. As far as transaction data reporting to The Financial Supervisory Authority of Norway is concerned, the only file type extension accepted is .zip, and the -zip extension is used both for encrypted (mandatory in the NO production environment) and un-encrypted (optional in the industry test environment).

2. Encryption, signing and eIDAS qualified trust service providers

a) What exactly is eIDAS?

Applies to: NO-006, NO-007, NO-008 and NO-010 file level error codes

eIDAS is basically an EU initiated regulation aimed at harmonizing the application of electronic identities (eIDs) and digital signatures held by individual citizens, companies and government institutions across Europe.

eIDAS seeks to enhance trust in electronic transactions in the EU's internal market by providing a common foundation for secure electronic interaction between citizens, businesses and public authorities cross-borders, in order to increase the effectiveness of public and private online services, electronic business and electronic commerce in the Union.

To concretise, in this regard, the eIDAS Regulation:

- Ensures that people and businesses can use their own national electronic identification schemes (eIDs) to access public services in other EU countries where eIDs are available.
- Creates an European internal market for electronic trust services – namely electronic signatures, electronic seals, time stamp, electronic delivery service and website authentication – by ensuring that they will work across borders and have the same legal status as traditional paper based processes.

The regulation will replace the current eSignatures Directive and any current inconsistencies in Digital Signature law across Europe. It was adopted by the General Affairs Council in July 2014, with regulations for trust services coming into force 1st July 2016. The mandatory mutual recognition of electronic identities (eIDs) will apply from mid-2018.

eIDAS covers authentication, signature seals, registered delivery services and time stamps.

b) What is an eIDAS qualified provider?

Applies to: NO-006, NO-007, NO-008 and NO-010 file level error codes

As far as reporting of transaction data according to MiFID art. 26 is concerned, an eIDAS qualified provider is basically any trust service provider qualified for issuing so-called **eSeals**.

The policy The Financial Supervisory Authority of Norway has established for the PKI solution used with MiFID art. 26 transaction reporting system, imposes on the submitting entities to use enterprise class certificates or so-called eSeals, issued by trust service providers qualified according to the eIDAS decree.

A responsibility for maintaining an updated list of which eIDAS qualified trust service providers are situated within each country of the EU/EØS area, and keeping such a list available to the public, rests on the national government agencies/supervisory authorities for electronic communication within the respective countries.

The eIDAS trusted list, containing the qualified trust service providers (TSPs) situated in Norway, is maintained by Norwegian Communication Authority (NKOM), https://eng.nkom.no/, whereas i.e. in the United Kingdom, the Department for Business, Innovation & Skills maintains and administer the equivalent list of TSPs situated in the UK.

c) How do I identity eIDAS qualified trust service providers situated within my country/region?

Applies to: NO-006, NO-007, NO-008 and NO-010 file level error codes

A so-called trust list, constituting a catalogue over eIDAS qualified trust service providers situated in each country, can be obtained from the respective national communication authorities.

According to the Norwegian Communication Authority, any eiDAS qualified trust service provider is, independent of the trust service providers country of situation, authorized/qualified to issue digital certificates (eSeals) to any legit company or organisation situated within the EU/EØS area.

The eIDAS project has, however, based on the respective eIDAS trust lists published by the communication authorities within the EU/EØS countries, made a couple of interactive tools available from the project website, that allows the public to search for eIDAS qualified trust service providers.

Relevant links (URLs):

eIDAS Trust list browser, https://webgate.ec.europa.eu/tl-browser/#/
eIDAS Map, https://www.eid.as/fileadmin/eidas-tsp-map/#/

d) How do I verify that I apply the correct procedure when encrypting and signing?

Applies to: NO-006, NO-007, NO-008 and NO-010 file level error codes

Even though Finanstilsynet's private encryption key is obviously not available to submitting entities doing internal testing of their encryption procedure, an end-to-end-test, including every step from encryption, through digital signing to signature verification and decryption, could very well be simulated in the submitting entity's own environment, using solely the keys incorporated in the submitting entities own X.509-formatted enterprise certificates. Or, to phrase it differently, **submitting entities doesn't really need Finanstilsynet's private key in order to verify that the encryption and signing procedure correctly is applied correctly.**

Before encrypted and digitally signed test files are submitted to the industry test environment, we highly recommend that the submitting entities internally verify the procedure they apply, using the GPGSM command pattern described below:

1. (Encrypt)

```
gpgsm --homedir <local keyring location> -–recipient <recipient key ID> --cipher-algo AES256 --
disable-policy-checks --disable-crl-checks --disable-ocsp --disable-trusted-cert-crl-check --output
TR_<SEIC>_<ORI>_<YYYYMMDD>_<RFSEQ>.ZIP.ENC   --encrypt
TR_<SEIC>_<ORI>_<YYYYMMDD>_<RFSEQ>.ZIP
```

2. (Sign)

```
gpgsm --homedir <local keyring location> --local-user <recipient key ID> --disable-policy-checks --
disable-crl-checks --disable-ocsp --output
TR_<SEIC>_<ORI>_<YYYYMMDD>_<RFSEQ>.ZIP.ENC.SIG --sign
TR_<SEIC>_<ORI>_<YYYYMMDD>_<RFSEQ>.ZIP.ENC
```

3. (Verify)

```
gpgsm --homedir <local keyring location> --disable-crl-checks –output

TR_<SEIC>_<ORI>_<YYYYMMDD>_<RFSEQ>.ZIP.ENC.SIG.UNSIGNED --verify

TR_<SEIC>_<ORI>_<YYYYMMDD>_<RFSEQ>.ZIP.ENC.SIG
```

4. (Decrypt)

```
gpgsm --homedir <local keyring location> --disable-crl-checks  --output
TR_<SEIC>_<ORI>_<YYYYMMDD>_<RFSEQ>.ZIP.ENC.SIG.UNSIGNED.DECRYPTED.ZIP --decrypt
TR_<SEIC>_<ORI>_<YYYYMMDD>_<RFSEQ>.ZIP.ENC.SIG.UNSIGNED
```

3. Transaction Report Validation

a) How does file validation and content validation differ?

File validation – verify compliance of the file with the XML schema (syntax of the whole file and specific transaction reports). If the file is not compliant, the whole file (all transactions included in the file) is rejected.

Content validation – a set of validation rules that are executed for each transaction report and verify the content of specific fields. Incorrect transaction reports are rejected, whereas correct transactions are processed in further steps. These validation rules include validations dependant on instrument reference data.

b) How does the 7 days grace period for transaction report validation work?

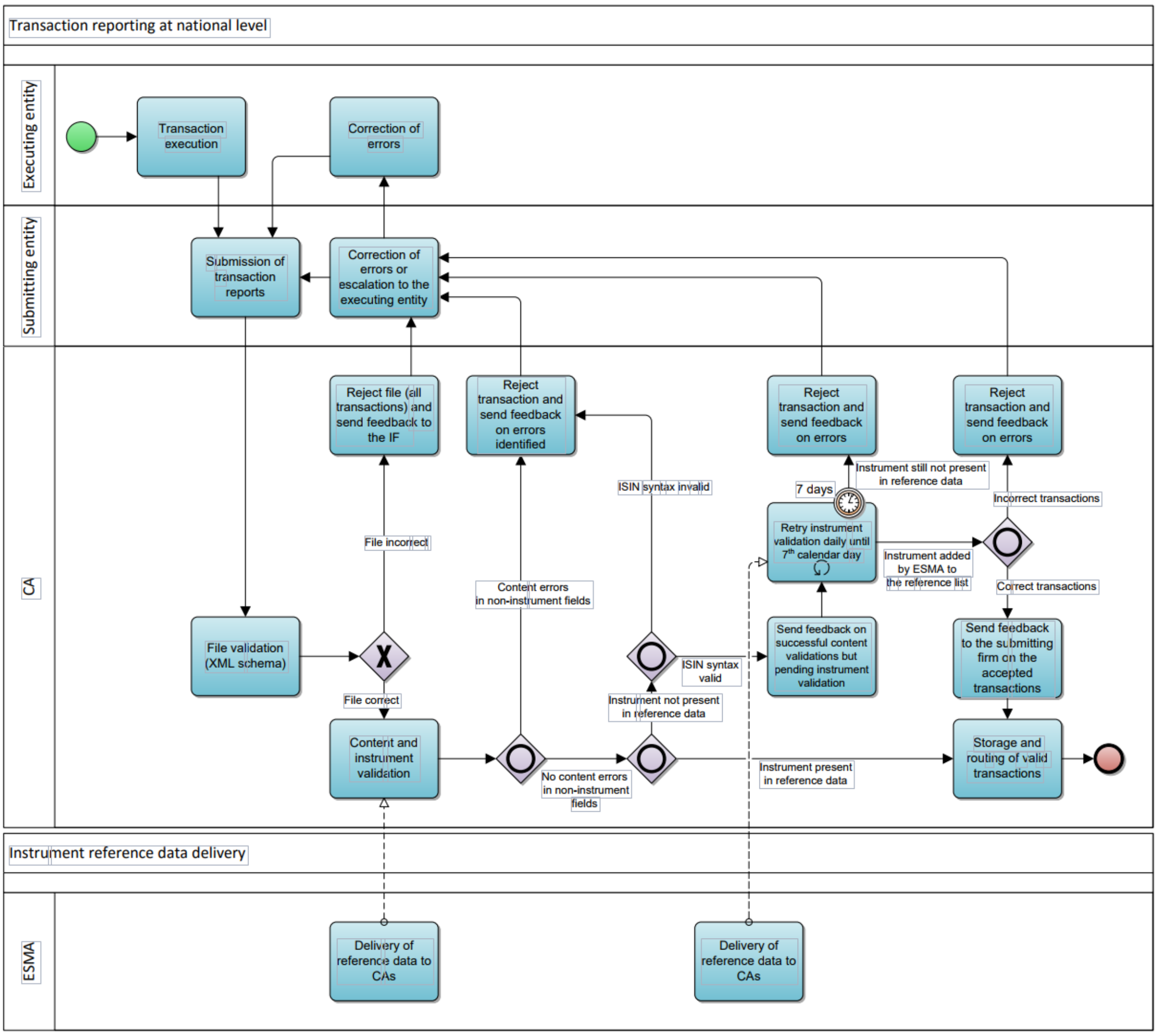

Transaction reporting at national level
Executing entity
Transaction execution
Correction of errors
Submitting entity
Submission of transaction reports
Correction of errors or escalation to the executing entity
CA
Reject file (all transactions) and send feedback to the IF
Reject transaction and send feedback on errors identified
Reject transaction and send feedback on errors
Reject transaction and send feedback on errors
Instrument still not present in reference data
7 days
ISIN syntax invalid
Incorrect transactions
Retry instrument validation daily until 7th calendar day
Instrument added by ESMA to the reference list
File incorrect
Content errors in non-instrument fields
Correct transactions
File validation (XML schema)
ISIN syntax valid
Send feedback on successful content validations but pending instrument validation
Send feedback to the submitting firm on the accepted transactions
File correct
Instrument not present in reference data
Content and instrument validation
No content errors in non-instrument fields
Instrument present in reference data
Storage and routing of valid transactions
Instrument reference data delivery
ESMA
Delivery of reference data to CAs
Delivery of reference data to CAs

Figure 1 - Transaction Report Validation

If the instrument/underlying referred in a reported transaction is missing in the reference data at the time the transaction report is received by the NCA, the following steps are undertaken:

i. the NCA will, through the status advice element in the corresponding feedback file, inform the submitting entity that the transaction is pending the instrument / underlying validation;

ii. the NCA will execute the instrument/underlying validation every day until the 7th calendar day after the report reception from the submitting entity;

iii. if the instrument/underlying reference data becomes present in the reference data on or before 7 calendar days have elapsed, and there are no content errors as a result of instrument validation, the transaction will be accepted. In such case, the submitting entity is notified of the acceptance through an updated status advice in the feedback file which is generated and made available to the submitting entity on a daily basis.

iv. If the instrument / underlying becomes present in the reference data on or before 7 calendar days have elapsed and there is a content error(s) as a result of instrument validation, the transaction will be rejected. In this case, the submitting entity is notified of the rejection through an updated status advice in the feedback file which is generated and made available to the submitting entity on a daily basis.

v. if after 7 calendar days the instrument / underlying is still not present in the reference data, the NCA will reject the transaction report.

4. File Error Codes and their respective meaning

a) What are the file validation error codes specific for Norway?

Additional to the common file level validation rules imposed by the TREM Interface specification, which applies for all countries in the EU/EØS, there are also more specific validation rules and corresponding error codes, which are applied only on a national level. The following controls/file level error codes are specific for Norway:

Control	Error code	Error message	Explanation and corrective action
File **name** syntax[1]	NOX-001	The file name has an illegal syntax.	Correct the filename so that it complies to the file name convention.
Invalid date	NOX-002	The date does not exist or is in the future.	Correct the date.
Duplicate	NOX-003	The file has already been submitted.	A file with identical values for SEIC, ORI, YYYYMMDD and RFSEQ in the file name has already been received by the TRS system. The duplicate file is ignored. If the most recently submitted version of a file contains new transaction reports (other than the ones submitted in the previous version of the file with the same name), submit these new transactions in a new report file with a different sequence number.
File extension	NOX-004	The file has an illegal extension.	Correct the filename so that it complies to the file name convention.

Control	Error code	Error message	Explanation and corrective action
File size	NOX-005	The file comprises more than 500 000 reports.	Split the report file into two or more report files.
Signature	NOX-006	The signature is not recognised as a valid signature	Check and correct the signature of the file. The signing key might be outdated or unknown to us. Also make sure the correct signature algorithm has been used.
Signer	NOX-007	The file has an unknown signer.	The signer is unknown to us. Check and correct the signature of the file.
Decryption	NOX-008	The file can't be decrypted.	Ensure that the file is properly encrypted. Check the encryption keys you are using or your encryption program. The encryption key might be outdated or unknown to us. Also make sure the correct hashing algorithm has been used.
Decompression	NOX-009	The file can't be decompressed.	Check that the compression method applied to the submitted file comply to the allowed compression method.
Signing	NOX-010	The file is not properly signed.	Ensure that the file is properly signed. Check the keys you are using. The key might be outdated or unknown to us.
SEIC	NOX-011	The SEIC is invalid.	Correct the SEIC in the file so that it is a valid legal entity identifier as defined in ISO 17442.
Recipient country	NOX-013	The recipient country, specified in the header, is incorrect.	Ensure that the recipient is the country code of the country of the CA to which the file is submitted.
Report file name	NOX-014	The report file name is incorrect.	Correct the name of the compressed report file so that it matches the name of the archive file and has the proper extension.
One report file per archive	NOX-015	The archive file does not contain only one report file.	Correct the contents of the archive file so that it contains only one report file.
Reporter allowed	NOX-016	The Submitting Entity is not authorized to report on behalf of the Executing Entity.	Ensure that it has been registered by the Executing Entity or the FSA that the Submitting Entity is authorised to report on behalf of the Executing Entity.

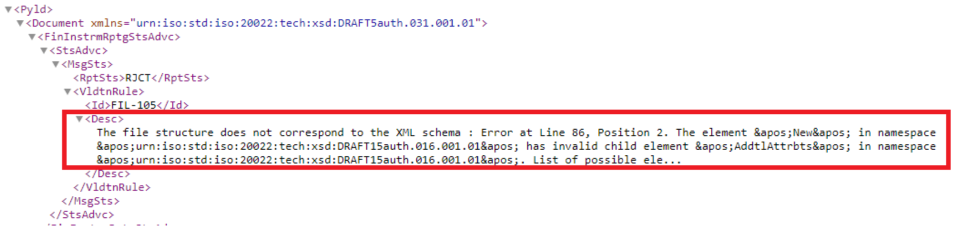
Control	Error code	Error message	Explanation and corrective action
SEIC	NOX-017	The SEIC is incorrect.	Correct the SEIC so that it is the LEI of the Submitting entity.
Submitting Entity in BAH	NOX-018	The Submitting entity, specified in the header, is incorrect.	Ensure that the Submitting Entity in the header, is the LEI of the Submitting Entity that submitted the file.
Submitting Entity in report	NOX-019	The Submitting Entity, specified in the report, is not the one that submitted the file.	Ensure that the Submitting Entity Identification code in the report, is the LEI of the Submitting Entity that submitted the file.

[1] This syntax control only applies to number of characters, separators, and numeric format used in the file name, and has nothing to do with syntax or structure of the actual file content.

b) Why are feedback files sometimes incomplete or truncated?

Why does the textual error description in the /BizData/Pyld/Document/FinInstrmRptgStsAdvc/StsAdvc/MsgSts/VldtnRule/ element in the feedback files we receive from the TRS system sometimes appear to be incomplete/truncated?

Applies to: FIL-105 file level error code



```
▼<Pyld>
  ▼<Document xmlns="urn:iso:std:iso:20022:tech:xsd:DRAFT5auth.031.001.01">
    ▼<FinInstrmRptgStsAdvc>
      ▼<StsAdvc>
        ▼<MsgSts>
           <RptSts>RJCT</RptSts>
          ▼<VldtnRule>
             <Id>FIL-105</Id>
            ▼<Desc>
               The file structure does not correspond to the XML schema : Error at Line 86, Position 2. The element &apos;New&apos; in namespace
               &apos;urn:iso:std:iso:20022:tech:xsd:DRAFT15auth.016.001.01&apos; has invalid child element &apos;AddtlAttrbts&apos; in namespace
               &apos;urn:iso:std:iso:20022:tech:xsd:DRAFT15auth.016.001.01&apos;. List of possible ele...
             </Desc>
           </VldtnRule>
         </MsgSts>
       </StsAdvc>
```

2 Truncated Error Description

When the textual error description in the /BizData/Pyld/Document/FinInstrmRptgStsAdvc/StsAdvc/MsgSts/VldtnRule/ element field appears as incomplete (truncated), this is simply due to a limitation imposed by the XML Schema Definition (XSD) that formally describes the elements and data types defined for usage within the transaction report status advice/feedback format. When a transaction report file is received from a submitting entity in the TRS system, one of the initial automated processing steps at Finanstilsynet's end is to verify whether the file received is structured according to the element type definitions and limitations imposed by the XSD.

This format validation routine relies on a XSD validation function, provided by one of the standard XML llibraries that comes with the MS .Net framework. In cases where the XML formatted transaction report file received is not structured in accordance with the XSD defined for the transaction report format, this XSD validation function will return a textual description with an account of the nature of the error. The length of the text string constituting this error description will, obviously, vary with the characteristics of the error in each case. However, since the transaction report status advice/feedback format XSD limits the length of the /BizData/Pyld/Document/FinInstrmRptgStsAdvc/StsAdvc/MsgSts/VldtnRule/ element in the feedback files to 350 chars, in cases when the textual error description returned by the MS .Net framework XSD validation function utilized by the TRS system exceeds 350 chars, only the first 350 chars of the error message will fit into the /BizData/Pyld/Document/FinInstrmRptgStsAdvc/StsAdvc/MsgSts/VldtnRule/ element of the feedback file uploaded to the submitting entities /Incoming folder on the SFTP server.

5. Transaction state reconciliation

a) **Is there a way to reconcile transaction state?**

From the 1. February 2019, Finanstilsynet has enabled a new feedback file for the Transaction Reporting System ("TRS"). All submitting entities ("SE") of TRS-data now have access to a folder named "StateOfTransactions" in their SFTP TRS file area.

The file names in the "StateOfTransactions" folder has the following format `shortnameofee-LE1COD36EVEQT5KFZZ64.json.zip`

For every executing entity ("EE") the SE has uploaded transactions on behalf of, the system will create a state file. The SE can only see the transactions of the EE that the SE itself has submitted.

The file is a JSON file compressed with zip. The JSON is structured as follows, and should be mostly self-explanatory:

```
{
    "comment": "This is an experimental feature! This file contains the state of all transactions, in the given date range below, for the executing entities of the submitting entity. Only transactions that are submitted by the submitting entity is included here; in other words, the executing entities could have transactions at other submitting entities, and those would not show up in this file. The intended use of this file is to provide an alternative method to reconcile state. File is updated at 01:00 every night, but this might change(!) you should rely on feedback files for continuous state.",
    "created": "2019-02-05T10:45:06.3809617+01:00",
    "fromdate": "2018-01-01",
    "todate": "2018-01-10",
    "executingentity": "7967017ERWSXZZA40T66",
    "totaltransactioncount": 10825,
    "statetypecount": {
    "Accepted": 7314,
    "Awaiting reference data": 50,
    "Cancelled": 3343,
    "Rejected": 118
    },
    "States": {
        "Accepted": [
        "5TTEDFZXA4000007587",
        "5TTEDFZXA4000007588",
        "5TTEDFZXA4000007589",
        "5TTEDFZXA4000007590",
        ...
        ],
        "Rejected": [
        "5TTEDFZXA4000012351",
        "5TTEDFZXA4000012352",
        "5TTEDFZXA4000015390",
        ...
        ],
        "Cancelled": [
        "5TTEDFZERS000012451",
        "5TTEDFZERS000012452",
        "5TTEDFZERS000015590",
        ...
        ],
        "Awaiting reference data": [
        "5TTEDEERD4000012451",
        "5TTEDEERD4000012452",
        "5TTEDEERD4000015590",
        ...
        ]
    }
}
```

The file is overwritten and updated around 01:00 CET every night. The file is not a part of the official TRS protocol, and implementations should not rely on state file data for correct operation