Pursuant to the National Market System Plan Governing the Consolidated Audit Trail (the “CAT NMS Plan”), the Operating Committee of CAT NMS, LLC (the “Operating Committee”) has authorized the publication of an excerpt discussion draft of v0.2 of the CAT Reporting Technical Specifications for Industry Members (the “Technical Specifications”) developed by the Plan Processor, Thesys CAT LLC (Thesys CAT). This discussion draft is an excerpt of the CAT Reporting Technical Specifications for Industry Members including only new sections that have not yet been published. The Operating Committee acknowledges that this draft is incomplete and does not reflect resolution of interpretive questions raised by the industry to date. The Operating Committee, however, believes it is important for the industry to play an active role in the development of the Technical Specifications and has therefore elected to share this excerpt so that industry comment may continue. As interpretive items are resolved, they will be incorporated into future versions of the Technical Specifications.

The Operating Committee has not approved or adopted the Technical Specifications for use by Industry Members; rather, the Operating Committee authorized the publication of the excerpt Technical Specifications to facilitate discussions with, and comments from, Industry Members regarding the Plan Processor’s proposed manner of reporting data to the Central Repository. The Operating Committee encourages Industry Members to provide comments on the draft Technical Specifications. Industry Members can send comments on the draft Technical Specifications to techspec@thesyscat.com.
# Table of Contents

11. SUBMISSION PROCESS .................................................................................................................. 5  
11.1. FILE AND DATA FORMATS .................................................................................................... 5  
11.1.1. FILE NAMES ....................................................................................................................... 5  
11.1.2. METADATA FILES ............................................................................................................. 6  
11.1.3. DATA FILES ...................................................................................................................... 8  
11.1.3.1. JSON SCHEMA ............................................................................................................... 8  
11.1.3.2. CSV CONVERSION ......................................................................................................... 11  
11.2. CONNECTIVITY ....................................................................................................................... 13  
11.3. TRANSPORT OPTIONS .......................................................................................................... 13  
11.3.1. FILE SIZE, ENCRYPTION, AND COMPRESSION .............................................................. 14  
11.3.2. SFTP UPLOAD PROCESS .................................................................................................. 15  
11.4. ACCESSING FEEDBACK INFORMATION .............................................................................. 16  
11.4.1. CAT FEEDBACK .................................................................................................................. 16  
11.5. CAT REPORTING HOURS ...................................................................................................... 16  
11.6. SECURITY .............................................................................................................................. 18  
11.6.1. DATA SECURITY STANDARDS ............................................................................................ 19  
12. FEEDBACK AND CORRECTIONS ............................................................................................... 20  
12.1. FEEDBACK FILES ................................................................................................................... 20  
12.2. FILE ACKNOWLEDGEMENT .................................................................................................. 22  
12.2.1. FILE ACKNOWLEDGEMENT FEEDBACK ...................................................................... 22  
12.3. BASIC FILE INTEGRITY ......................................................................................................... 23  
12.3.1. BASIC FILE INTEGRITY CHECKS .................................................................................... 23  
12.3.2. BASIC FILE INTEGRITY FEEDBACK .............................................................................. 24  
12.4. ORDER EVENTS FILES ......................................................................................................... 25  
12.4.1. ORDER EVENT FEEDBACK .............................................................................................. 26  
12.5. CORRECTIONS ......................................................................................................................... 28  
12.5.1. REPAIR RECORD .............................................................................................................. 28  
12.5.1.1. DELETE RECORD ........................................................................................................... 29  
12.5.1.2. CORRECTION RECORD ................................................................................................. 29
11. Submission Process

In this section, information is provided regarding how to format submission files, submit to CAT (including a general data flow overview), the registration process, network and transport options, and CAT access and reporting hours.

11.1. File and Data Formats

All files sent from the reporter (or the third-party submitting agent for the reporter) must be compressed, encrypted, and signed. However, the information in this section assumes, for the most part, that the file being described is the raw, unencrypted data (after being verified, decrypted, and uncompressed).

All files submitted on a particular date must have a unique file name, as defined in 11.1.1. The mechanism used for uploading files will prevent duplicate file names from being accepted into the CAT system.

Archive files are not allowed to be submitted into the CAT system. Each set of records must be submitted as an individual compressed, encrypted, and signed file. For example: Do NOT zip, tar, or 7z all of the submitted files into one big file. The files will be individually compressed, encrypted, and submitted.

All data elements are submitted using ISO-8859-1 encoding. This is a one-byte-per-character encoding, with possible values in the range of [0, 255]. This encoding has the characteristic that the encoding character definitions are the same as the first 256 code points of UTF-8. However, only fully defined values will be accepted.

According to the encoding specification, byte values in the ranges [0, 31] and [127, 159] are undefined. As a result, any record submitted with character values in those ranges will be rejected as invalid. In cases where data is echoed back in feedback files, invalid characters will be translated to a 3-character octal value, preceded by a backslash.

11.1.1. File Names

Files are to be named in the following manner: <CAT Reporter ID>_<Date>_[<Group>_]<File Kind>_<File Number>_<Extension>[.<Compression Extension>].<Encryption Extension>, where

- CAT Reporter ID is the unique ID assigned to the reporter by CAT
• Date is the business date for all events in the file in YYYYMMDD format - not the date the file was generated or reported
• Group is an optional reporter-defined string. It must either be missing, or composed of up to 20 alphanumeric characters. The field exists solely for reporters’ convenience, e.g. Industry Members may put Reporter IMID in here. Other than file name validation, it is ignored by the CAT processor
• File Kind is “OrderEvents” for Industry Members
• File Number is the sequence number of this file, 6-digits long, left-padded with zeros The tuple (CAT Reporter ID, Date, File Kind, File Number) must be unique. The File Number determines the order that a file will be processed within a File Kind.
• Extension is the extension, representing the format of the data inside: json, csv
• Compression Extension is the extension representing the compression used to compress the data file: gz, bz2, xz, zip. It is only needed if compression is done outside of the encryption process. If your OpenPGP tool handles compression, Compression Extension should be left off
• Encryption Extension is the extension indicating that the file is encrypted and must be either .gpg or .pgp.

11.1.2. Metadata Files

For each data file that is uploaded to CAT, an associated metadata file must also be uploaded. The metadata file has the same base name as the data file, with .json or .csv extension replaced by .meta. The metadata file does not need to be compressed, but must still be encrypted and signed like the data file. The metadata file is in JSON format, and contains:

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>Message Type</td>
<td>META</td>
</tr>
<tr>
<td>businessDate</td>
<td>Date</td>
<td>The business date for the data contained within the data file - a single data file must contain data for a single calendar date.</td>
</tr>
<tr>
<td>reporter</td>
<td>Reporter ID</td>
<td>The CAT-assigned ID for the entity required to report data to CAT.</td>
</tr>
<tr>
<td>Field</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>submitter</td>
<td>Reporter ID</td>
<td>The CAT-assigned ID for the entity submitting data on behalf of the reporter. The submitting entity must have been previously authorized to submit data on behalf of the reporter. If the reporter submits data on their own behalf, then the submitter would be the same as the reporter.</td>
</tr>
<tr>
<td>fileVersion</td>
<td>Version</td>
<td>A version number for the schema file used to encode and format this file. The schema file will be inferred from the File Kind of the file.</td>
</tr>
<tr>
<td>recordCount</td>
<td>Unsigned</td>
<td>The number of new-line delimited records in the data file</td>
</tr>
<tr>
<td>rawHash</td>
<td>Alphanumeric (64)</td>
<td>SHA256 of the raw data file. This field and/or compressedHash must be provided.</td>
</tr>
<tr>
<td>compressedHash</td>
<td>Alphanumeric (64)</td>
<td>SHA256 of the compressed data file. This field and/or rawHash must be provided.</td>
</tr>
<tr>
<td>encryptedByteCount</td>
<td>Unsigned</td>
<td>The total number of bytes in the data file, after having been compressed and encrypted.</td>
</tr>
<tr>
<td>encryptedHash</td>
<td>Alphanumeric (64)</td>
<td>SHA256 of the encrypted data file</td>
</tr>
<tr>
<td>symmetricKey</td>
<td>Alphanumeric (64)</td>
<td>The symmetric key or passphrase used to encrypt the file.</td>
</tr>
<tr>
<td>isKindDone</td>
<td>Boolean</td>
<td>Used to indicate the last file for this File Kind (OrderEvents) for the submitter/reporter on the businessDate. Any file submitted with isKindDone=true should be the last file submitted for the File Kind and should have the highest File Number. It defaults to false.</td>
</tr>
</tbody>
</table>

Unrecognized field names are ignored. The metadata file must contain exactly one JSON object, and it may contain new-lines as whitespace.
The hashes are to be submitted as 64-character hexadecimal string encodings of the hash value.

11.1.3. Data Files

All data files are either new-line delimited JSON objects, or new-line delimited CSV records. This means that for JSON, there is no top level object. Instead, the file acts as the top level container for each object. Each object is a normal JSON object, separated with a new-line (ASCII decimal 10, hex 0A). For CSV files, each record’s fields are separated with a comma (ASCII decimal 44, hex 2C).

Each JSON object is terminated by a new-line, but the data in the object itself must not include new-lines. Specifically, each line in the file must contain exactly one complete record, no matter whether the submission format is JSON or CSV. In either case, the total maximum length of any line is 4095 bytes. The examples in the document include new-lines between elements for readability.

11.1.3.1. JSON Schema

JSON schema files for each record type will be provided on the CAT public website. Reporters will be able to download and use these schemas to format and validate their CSV and JSON data files prior to submission. These schemas will also allow reporters to translate their data files from JSON to CSV or from CSV to JSON formats, as desired.

The schema files will be maintained by the CAT Processor and will be versioned as the message specifications change. The meta files submitted to CAT will contain a version identifier specifying which version of the schema the associated reference or order data was formatted in accordance with. This will allow the CAT system to perform a basic initial formatting validation of all submitted data.

Provided here is an abbreviated example of a JSON schema containing only part of the equity New Order event and a couple definitions for Choice fields:

```json
{
    "description": "CAT schema for equity new order event",
    "version": "0.2",
    "eventDefinitions": [
        
        "eventName": "New Order",
        "fields": [
            
        ]
    ]
}
```
<table>
<thead>
<tr>
<th></th>
<th>JSONDataType</th>
<th>name</th>
<th>dataType</th>
<th>position</th>
<th>required</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>String</td>
<td>type</td>
<td>Choice</td>
<td>0</td>
<td>Required</td>
</tr>
<tr>
<td>2</td>
<td>String</td>
<td>reporterIMID</td>
<td>Industry Member ID</td>
<td>1</td>
<td>Required</td>
</tr>
<tr>
<td>3</td>
<td>[ &quot;Number&quot;, &quot;String&quot; ]</td>
<td>eventTimestamp</td>
<td>Timestamp</td>
<td>2</td>
<td>Required</td>
</tr>
<tr>
<td>4</td>
<td>String</td>
<td>symbol</td>
<td>Symbol</td>
<td>3</td>
<td>Required</td>
</tr>
<tr>
<td>5</td>
<td>String</td>
<td>orderID</td>
<td>Text</td>
<td>4</td>
<td>Required</td>
</tr>
<tr>
<td>6</td>
<td>String</td>
<td>memberTypeCode</td>
<td>Choice</td>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>
"required": "Required"
}
...
]
}
"choices": {
  "memberTypeCode": [ "A", "C", "E", "F", "M", "N", "X" ]
}
}

Note that the file is not a typical "JSON schema" but a schema describing the reportable events in JSON.

The field "JSONDataType" indicates the underlying JSON data type.

The field "dataType" is the actual type, as indicated in this specification, with some further restrictions over the underlying JSON data type.

The field "name" is the JSON field name. The "name" is also used as a lookup key to find valid values for a field of dataType "Choice."

Each field of dataType "Choice" will contain a corresponding entry in the "choices" object, which contains the list of valid choices. The key is the value in the name field. If the name field contains a "." (period), then the value is part of a nested JSON object and the key will be the trailing name. For example, if the field has a Choice field with the name "buyDetails.side" then the field "buyDetails" contains a JSON object, with a member named "side" and "side" would be used as the key to lookup the valid choices for the element.

The field "position" is the 0-based index where this field would be expected in a CSV version of the data.

The field "required" indicates whether the field is "Required," "Conditional," or "Optional." If submitting in JSON, any conditional or optional field that is not provided must be omitted. If submitting in CSV, and conditional or optional field that it not provided must be an empty column (i.e., in the following example position 2 is considered to be omitted: zero,1,,three).

Note that the Timestamp data type has two possible representations, so the JSONDataType is an array of choices: String for a formatted string and Number for nanoseconds since the epoch.
11.1.3.2. CSV Conversion

The JSON schema defines valid data types, and mappings between JSON and CSV. Note that schemas can change, and the authoritative schemas will be available on the CAT website. For this discussion, assume the following schema for the equity New Order event:

```json
{
    "eventName": "Trade Break",
    "fields": [
        {
            "name": "type",
            "dataType": "Message Type",
            "JSONDataType": "String",
            "required": "Required",
            "defaultValue": "METB",
            "position": "0"
        },
        {
            "name": "reporterIMID",
            "dataType": "Industry Member ID",
            "JSONDataType": "String",
            "required": "Required",
            "position": "1"
        },
        {
            "name": "eventTimestamp",
            "dataType": "Timestamp",
            "JSONDataType": "string,Number",
            "required": "Required",
            "position": "2"
        },
        {
            "name": "sequenceNumber",
            "dataType": "Unsigned",
            "JSONDataType": "Number",
            "required": "Required",
            "position": "3"
        },
        {
            "name": "symbol",
            "dataType": "Symbol",
            "JSONDataType": "String",
            "required": "Required",
```
"position": "4"
},
{
    "name": "tradeDate",
    "dataType": "Date",
    "JSONDataType": "String",
    "required": "Required",
    "position": "5"
},
{
    "name": "originalTradeID",
    "dataType": "Text",
    "JSONDataType": "String",
    "required": "Required",
    "maxLength": 40,
    "position": "6"
},
{
    "name": "priorCorrectionDate",
    "dataType": "Date",
    "JSONDataType": "String",
    "required": "Conditional",
    "position": "7"
},
{
    "name": "priorTradeID",
    "dataType": "Text",
    "JSONDataType": "String",
    "required": "Conditional",
    "maxLength": 40,
    "position": "8"
},
{
    "name": "quantity",
    "dataType": "Number",
    "JSONDataType": "Number",
    "required": "Conditional",
    "position": "9"
},
{
    "name": "initiator",
Below is a sample event based on the JSON representation:

```json
{
    "type": "METB",
    "reporterIMID": "MPID",
    "eventTimestamp": "20170901T120102.123456",
    "symbol": "XYZ",
    "tradeDate": "20170831",
    "originalTradeID": "T12345",
    "initiator": "Firm"
}
```

The corresponding CSV would be:

```
METB,MPID,20170901T120102.123456,,XYZ,20170831,T12345,,,,Firm,
```

11.2. Connectivity

11.3. Transport Options

Reporters may use different mechanisms (SFTP or the Reporter Web Portal) to send/obtain different types of information to/from CAT.

Basic types of CAT information:
1) Submissions (e.g. initial submission of files of order events, resubmission of files that were previously rejected, and corrections or deletions to previously accepted records; 2) Feedback (e.g. upload status, rejections, and reporting statistics); and 3) Administrative information.

<table>
<thead>
<tr>
<th>Access Method</th>
<th>Category</th>
<th>SFTP</th>
<th>Reporter Web Portal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Submission of Order Events</td>
<td>Submission</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Resubmission of Rejected Files/Records and Correction</td>
<td>Submission</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>File Status Retrieval</td>
<td>Feedback</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Report Statistics</td>
<td>Feedback</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>System Status</td>
<td>Administrative</td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>

SFTP submission allows file(s) only. If the submission is via the Reporter Web Portal, it may be sent by typing the information directly into the Web page or by submitting small files. The file size allowed via Reporter Web Portal is limited to 1GB before compression.

11.3.1. File Size, Encryption, and Compression

Any files transmitted via SFTP or Reporter Web Portal must be 1) compressed AND 2) encrypted AND 3) signed, for the purpose of security and efficient network usage. The following compression algorithms will be allowed:

- ZIP (extension: .zip)
- Gzip (extension: .gz)
- BZip2 (extension: .bz2)
- LZMA2 (extension: .xz)

The size of files uploaded via SFTP is limited to 100GB per file, before compression and encryption. Resumable file upload is enabled for SFTP. The size limit for the Reporter Web Portal is 1GB per file, before compression and encryption.
Submission of data to CAT requires a number of supporting requirements that include security, confidentiality, authentication, and provenance of the data submitted. To achieve this the standard capabilities of PGP will be utilized and will require encrypting, signing, and secure hashing of data. While these steps must be followed, submitters have a choice of compression algorithms, passphrase selection and length, as well as what tools they use on their platform. When encrypting a metadata file, the CAT public encryption key must be used. This public key will be made available to reporters. The file may also be encrypted with the public key of the submitter and the public key of the reporter. Encrypted files can only be decrypted by the private key corresponding with the public key used to encrypt the file; including the optional public keys of Submitter and Reporter ensures that private key holders of CAT, Reporter, and Submitter can decrypt the encrypted file.

Public / Private keys should be RSA with a bit length of at least 2048 (minimum and recommended), and up to 4096. Interested parties can read more at https://www.gnupg.org/faq/gnupg-faq.html sections 11.3-4 are directly relevant. The cipher algorithm must be AES-256 (recommended) or higher. When encrypting data files, the submitter may also sign the file with their private key. Metadata files must always be signed in cleartext, and then encrypted. All files must be compressed before, or with the encryption process. The digest algorithm used to sign the file must be SHA256. The signature should also be part of the encrypted file (i.e. no detached signatures). Once a file has been encrypted and/or signed, the extensions .pgp, .gpg, or .asc must be appended to the end of the filename; OpenPGP compliant tools will do this for you.

11.3.2. SFTP Upload Process

Each file that is uploaded must follow these basic steps.

1. Upload the data file and the metadata file into the upload/transit directory, which will be at the base of the home directory for each sftp account.

2. Upon successful upload of both files, move both files into the upload/complete directory. Note that both files should be uploaded before moving either to the complete directory.

A file should never be directly uploaded into the upload/complete directory, as once a file has appeared there it is assumed to be the desired complete file submission. Only files that have a .pgp or .gpg extension will be processed.

The CAT processor will remove files from the upload/complete directory, though the timing of removal is unspecified. The submitter should never attempt to delete files from the
11.4. Accessing Feedback Information

11.4.1. CAT Feedback

Multiple types of feedback will be provided to reporters through various mechanisms, dependent upon the type of feedback provided.

- **Receipts** - receipt of a file or the arrival of a file at the next stage of processing is provided in a feedback file. A File Acknowledgement Feedback File will be provided when each file is first received. A separate file will be generated when the file reaches each stage of processing. Processing stages (and thus feedback file types) vary based on the type of file being processed. Feedback files will be available via sftp. Receipt and feedback information will also be available via the Reporter Web Portal.

- **Failure Reports** - If records in a file are rejected, if an entire file is rejected, or if there are warnings generated by a file, the feedback file will detail which records were rejected, why they were rejected, and at which stage of processing they were rejected. Note that in the case of an entire file rejection, only a subset of the records will be provided in the feedback. This is because once an error limit is reached (10% of the records), the entire file is rejected.

- **File submission status** - current processing status (whether a file has been received, which stage of processing a file is in, etc.) will be made available via the Reporter Web Portal.

11.5. CAT Reporting Hours

**Submission of Order Events**

Pursuant to SEC Rule 613, the CAT NMS Plan requires reporters to record order events contemporaneously with the actual transactions themselves. Realtime reporting to CAT is not required. Data may be bulk uploaded at the end of the trading day, or may be broken into multiple batches and uploaded in pieces throughout the day. However, all reportable events for one trading day (**before midnight Eastern Time**) must be reported to CAT by 8:00 AM Eastern Time on the next trading day.
*Please note the definition of Trading Day is currently being reviewed by the Operating Committee*

CAT accepts submissions (via SFTP and Reporter Web Portal) 24 hours per day, 7 days per week, other than during announced scheduled maintenance. Events that occurred during a particular Trading Day may be reported anytime between the time the event occurred and the reporting deadline, which is 8:00 AM Eastern Time on the following Trading day. Reports received after the deadline will be marked late by CAT.

The table below gives some examples of the reporting deadline.

<table>
<thead>
<tr>
<th>Event Occurs</th>
<th>Holiday</th>
<th>Report Due to CAT (T+1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monday 14:20 PM ET</td>
<td>N/A</td>
<td>Tuesday 8:00 AM ET</td>
</tr>
<tr>
<td>Monday 23:40 PM ET</td>
<td>N/A</td>
<td>Tuesday 8:00 AM ET</td>
</tr>
<tr>
<td>Friday 11:00 AM ET</td>
<td>N/A</td>
<td>Monday 8:00 AM ET</td>
</tr>
<tr>
<td>Friday 16:02 PM ET</td>
<td>N/A</td>
<td>Monday 8:00 AM ET</td>
</tr>
<tr>
<td>Friday 16:02 PM ET</td>
<td>The Following Monday</td>
<td>Tuesday 8:00 AM ET</td>
</tr>
<tr>
<td>Wednesday 15:00 PM ET</td>
<td>Thursday, Friday is half day</td>
<td>Friday 8:00 AM ET</td>
</tr>
</tbody>
</table>

**Deadline of Rejection Repair**

Rejections will be provided to Reporters in the following order:

- File Format Validation Error
- Syntax and Semantics Error
- Context Issues

Once rejections are available, repairs can be made immediately.

In order to comply with the rule, all Rejections that require repair should be repaired before **8AM Eastern Time on T+3** (transaction date + three Trading Days). Repairs received after the standard repair window will be classified as late.

If corrections are not received by **8AM Eastern Time T+5** (transaction date + five Trading Days), Participants’ regulatory staff and the SEC will be notified. The Plan Processor shall
notify the Participants’ regulatory staff and the SEC as to how corrections submitted after T+5 will be re-processed. The Operating Committee will be involved with decisions on how to reprocess the data.

<table>
<thead>
<tr>
<th>Event Occurs</th>
<th>Holiday</th>
<th>Initial Report Due (T+1)</th>
<th>Standard Repair Window (T+3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monday 14:20 PM ET</td>
<td>N/A</td>
<td>Tuesday 8:00 AM ET</td>
<td>Thursday 8:00 AM ET</td>
</tr>
<tr>
<td>Monday 23:40 PM ET</td>
<td>N/A</td>
<td>Tuesday 8:00 AM ET</td>
<td>Thursday 8:00 AM ET</td>
</tr>
<tr>
<td>Friday 11:00 AM ET</td>
<td>N/A</td>
<td>Monday 8:00 AM ET</td>
<td>Wednesday 8:00 AM ET</td>
</tr>
<tr>
<td>Friday 16:02 PM ET</td>
<td>N/A</td>
<td>Monday 8:00 AM ET</td>
<td>Wednesday 8:00 AM ET</td>
</tr>
<tr>
<td>Friday 16:02 PM ET</td>
<td>Next Monday</td>
<td>Tuesday 8:00 AM ET</td>
<td>Thursday 8:00 AM ET</td>
</tr>
<tr>
<td>Wednesday 15:00 PM ET</td>
<td>Thursday, Friday is half day</td>
<td>Friday 8:00 AM ET</td>
<td>Tuesday 8:00 AM ET</td>
</tr>
</tbody>
</table>

**Deadline for Corrections and Deletions**

Sometimes a reporter will have occasion to correct a report that may have passed all data validation and integrity checks. All such corrections must be submitted within the same three day timeframe as provided for record repairs. Specifically, reporters will be provided the same T+3 window for submitting timely corrections to data.

11.6. **Security**

In recognition of the importance of security, CAT has strict mechanisms for security controls. This section describes the CAT security overview and data security standards, to make sure the data is secured both during in transit and while stored in CAT.

Submission to CAT requires a valid user ID and password. Reporters must obtain a master user ID and password combination during the CAT registration process.

Reporters will also be assigned a multi-factor authentication (MFA) token during the registration process. An MFA token is required for authentication when accessing SFTP and Reporter Web Portal.
Order Submission or Correction via SFTP or Reporter Web Portal must be compressed and encrypted. All communication with the web portal is done through secure HTTPS, and all such activity is captured in system logs.

Reporters are not allowed to view or download the actual data files from SFTP or Reporter Web Portal after submission. Data will be deleted from SFTP in a predefined time window, after it has been successfully processed and loaded into CAT data store.

### 11.6.1. Data Security Standards

In addition to data transport security provided by TLS and SSH, CAT requires that the data be secure at rest. To achieve this, CAT recommends one of the supported, industry standard tools for encryption to be used. PGP, OpenPGP, and GPG (GNU Privacy Guard), in addition to OpenSSL may be used for both symmetric and asymmetric encryption and decryption. Asymmetric encryption will be accomplished by the reporter using the reporter's private key and the CAT public key. Upon retrieval by CAT, the CAT private key will be used to decrypt the files. Should the reporter desire to be able to decrypt the data at another time, the reporter should encrypt with both the reporter's public key, as well as the CAT public key.

Automated systems are an anticipated component of CAT submission, error retrieval, and status monitoring, and automated access is permitted in limited roles.

CAT provides restricted automated (role based) sub-accounts with restricted access. Please note that automated access accounts must be associated with a person. These accounts will be permitted to use public security keys from their registered location. Public security keys will be registered via the Web portal, and associated with the restricted functional account. Should a Participant desire HSM support, most HSM devices allow public key extraction using commonly available tools from OpenSSL and OpenSSH, utilizing the `pkcs12`, `ssh-keygen` modules.
12. Feedback and Corrections

This section describes the procedures for obtaining feedback and how to submit corrections, including different types of feedback messages, data elements, and formats of the correction reports. After data submission, CAT will conduct data validations, provide feedback to submitters and reporters, and allow corrections to be submitted.

Feedback will be made available via the Reporter Web Portal, both in the form of a user-centric webpage and a programmable API.

For descriptions in this section, we will consider a file to be successfully decrypted if and only if all of the decryption, decompression, and signature validation steps succeed.

12.1. Feedback Files

Receipts will be available as soon as they are generated in the form of a feedback file, accessible under a subdirectory of download/feedback in both the submitter's and reporter's home directory on the feedback sftp server. They will also be made available via the Reporter Web Portal. Note that the IP address of the sftp server where files are uploaded may differ from the IP address of the sftp server where receipts and failures are placed.

The exact subdirectory on the sftp server where feedback files may be found is:

```
download/feedback/YYYYMMDD/CATReporterId/
```

Where YYYYMMDD is the business date of the events in the submitted file.

The format of the feedback file is a JSON document. Each member of the top-level JSON object corresponds to the fields in the receipt. If there are failures or warnings, they will be available in the failures array on the main JSON object. It is possible for a record to have multiple failures, which will be represented by multiple JSON objects within the failures array with the same record indexes.

If any of the failures correspond to specific records, a failure file will also be generated in the download/failures directory in the home directory of both the submitter and reporter. The filename of these records will be in the feedback JSON object under the key failureFile. The failure file will be populated with Correction Records containing the original rejected records and record index, to make corrections easier to submit (as described in the Corrections section). The record index is the offset from the beginning of the file to the first byte of the record.
Feedback files will have the same base name as the submitted file. The file name will be appended with an extension describing the feedback type and .pgp. It will be compressed, encrypted, and signed. Compression will be based on the compression preferences associated with the public keys used to encrypt the feedback file. The keys used will be the submitter’s key, reporter’s key (if different), and the CAT public key. It will be signed with the CAT private key. For example, if a file was submitted from CAT Reporter “MYID”, with the following name:

MYID_20170101_OrderEvents_000123.csv.gz.pgp

The following would be the filename for the acknowledgement feedback file:

MYID_20170101_OrderEvents_000123.ack.pgp

Filnames are considered unique using the base name of the file (i.e., after removing all suffixes). Thus, trying to upload files that differ only in extension would be considered an error for uploading files with duplicate filenames.

<table>
<thead>
<tr>
<th>Filename</th>
<th>Basename - used for comparison</th>
</tr>
</thead>
<tbody>
<tr>
<td>MYID_20170101_OrderEvents_000123.csv.gz.pgp</td>
<td>MYID_20170101_OrderEvents_000123</td>
</tr>
<tr>
<td>MYID_20170101_OrderEvents_000123.csv.bz2.pgp</td>
<td>MYID_20170101_OrderEvents_000123</td>
</tr>
<tr>
<td>MYID_20170101_OrderEvents_000123.json.gz.pgp</td>
<td>MYID_20170101_OrderEvents_000123</td>
</tr>
</tbody>
</table>

If multiple files are submitted with the same base name, but with different format or compression extensions, then a _<N> will be applied to the base name of the second and subsequent feedback file names, where N will be the iteration of the feedback file.

If multiple files are submitted with the same base name or CAT needs to provide feedback when reprocessing a file feedback, an _<N> will be applied to the base name of the feedback files to avoid overwrite of any feedback files contained at that time in the download directories.

Feedback files will be generated as each data file goes through various stages of processing. Some stages will complete almost immediately, generating feedback shortly after the file has been uploaded; other stages may take significant time, meaning that feedback could be delayed (either due to queuing behind other jobs, or because certain data elements are required to advance).

The minimum retention time for feedback files on the sftp server is 10 business days; after that time they may be removed from the server. Feedback will continue to be available after that time via the Reporter Web Portal.
Failure codes are listed in Appendix B.

12.2. File Acknowledgement

A receipt of acknowledgement will be generated for each file and metadata pair that appears in the upload/complete directory or uploaded via the Reporter Web Portal.

The data and metadata file should be moved into the upload/complete directory within a reasonable timeframe of each other. If a file is rejected (e.g., because the filename is not in the correct format or there is a timeout without seeing the associated data/metadata file), the receipt will contain a status of Failure and one or more failure codes with descriptions.

Files will be removed from the upload/complete directory after they are processed, though they may not be removed immediately after processing.

12.2.1. File Acknowledgement Feedback

An acknowledgement feedback file will have a .ack extension and will contain the following fields:

- CAT Submitter ID (as determined from sftp or web portal username)
- CAT Reporter ID (as determined from filename, if available)
- Timestamp of Receipt - timestamp file is received and receipt is generated
- Business Date (as determined from filename, if available)
- File Name
- Status - Success or Failure
- Failures - a list of failure codes and descriptions
  - Severity - WARNING or ERROR
  - Failure Code
  - Failure Description (human readable, not intended to be processed)

The following is an example JSON object for a successful file acknowledgement:

```json
{
    "submitter": "SUBID",
    "reporter": "MYID",
    "receiptTimestamp": "20170307T153552.000001089",
    "businessDate": 20170307,
    "fileName": "MYID_20170307_OrderEvents_000123.json.gz.pgp",

```
"status": "Success"
}

The following is an example JSON object for an unsuccessful file acknowledgement:
{
  "submitter": "SUBID",
  "reporter": "MYID",
  "receiptTimestamp": "20170307T153552.000001089",
  "businessDate": 20170307,
  "fileName": "MYID_20170307_OrderEvents_000123.json.gz.pgp",
  "status": "Failure",
  "failures": [
    {
      "severity": "ERROR",
      "code": "FILE.TIMEOUT.001",
      "desc": "Time out waiting for meta file."
    }
  ]
}

12.3. Basic File Integrity

When both the data file and associated metadata file have been received, basic validation will begin. If the metadata cannot be decrypted, a failure will be generated, and no further attempt will be made to process the file until a valid metadata file is uploaded.

12.3.1. Basic File Integrity Checks

The values contained in the metadata file will be checked against properties of the corresponding data file. The following properties will be checked:

- **Matching Date** - the date part of the filename must match the metadata Business Date
- **Submitter** - metadata CAT Submitter ID must be the same as actual submitter (as determined from sftp or web portal username)
- **Reporter** - the CAT Reporter ID part of the filename must match the metadata CAT Reporter ID
- **Encrypted Size** - metadata Encrypted Byte Count must equal size of encrypted, received file
• **Encrypted Hash** - metadata Encrypted Hash must equal computed SHA256 of encrypted, received file
• **Decryption** - the data file must be successfully decrypted
• **Compressed Hash** - computed SHA256 must equal metadata Compressed Hash, if provided
• **Data Hash** - computed SHA256 must equal metadata Raw Hash, if provided

One or both of the Compressed Hash and Data Hash must be provided. If neither are provided, then the file will be rejected.

Note that all data elements in the metadata file are validated during this stage except Record Count, which will be validated when the file is actually processed.

### 12.3.2. Basic File Integrity Feedback

A basic file integrity feedback file will have a `.integrity` extension and will contain the following fields:

- CAT Submitter ID (as determined from sftp or web portal username)
- CAT Reporter ID (as determined from filename)
- Timestamp of Receipt - timestamp when receipt was generated
- Business Date
- Status - Success or Failure
- Failures - a list of failure codes and descriptions
  - Severity - WARNING or ERROR
  - Failure Code
  - Failure Description (human readable, not intended to be processed)

The following is an example JSON object for a successful integrity check:

```json
{
  "submitter": "SUBID",
  "reporter": "MYID",
  "receiptTimestamp": "20170307T153552.000001089",
  "businessDate": 20170307,
  "status": "Success"
}
```

The following is an example JSON object for an unsuccessful integrity check:

```json
{
  "submitter": "SUBID",
  "reporter": "MYID",
  "receiptTimestamp": "20170307T153552.000001089",
```

---

THIS IS A DRAFT DOCUMENT - FOR DISCUSSION PURPOSES ONLY
NOT FINAL - PLEASE DO NOT USE FOR CODING
NOT APPROVED BY OPERATING COMMITTEE OF CAT NMS, LLC
"businessDate": 20170307,
"status": "Failure",
"failures": [
{
   "severity": "ERROR",
   "code": "INT.META.001",
   "desc": "Encrypted SHA does not match SHA in meta file."
}
]

12.4. Order Events Files

Order Event files are composed of many different types of records. Any record determined to be malformed or otherwise invalid will be rejected as a failure.

- If more than 10 percent of the records in the file are rejected, the entire file will be rejected.
- If the number of records in the file does not match the Record Count in the metadata file, the entire file will be rejected.
- If an Order Event file contains anything other than expected order event messages, the entire file will be rejected.

Each field of the order event will be checked and validated, resulting in one of three states for the record: success, error, or warning. An error will prevent the record from being processed. A warning will not prevent the record from being processed. Depending on the type of warning, the record may be processed and ignored, or processed and applied to the data set.

For example, there are occasions where symbols are “delisted” late and may already have been referenced by some reporters (most likely in stage two). CAT allows incremental uploads throughout the day. Thus, their order event reports may contain opens and/or cancels for those symbols. Instead of rejecting these records, CAT will generate warnings for benign order actions and silently ignore them. Execution events for such symbols, however, will generate errors.

There may be cases where the system has a preferred method for reporting some order events but will accept other methods (especially if there is a change or transition to something new). In such cases where the preferred method is detectable, but not detrimental, a warning may be generated to inform the reporter what is happening, but the record will still be accepted and processed by the system.
12.4.1. Order Event Feedback

The act of processing order events has multiple stages: ingestion and linkage discovery.

During the ingestion phase, each record will be checked for proper formatting (JSON field names and values, CSV values in proper columns, etc) and data contents. The defined JSON schemas for each record type will be used to validate every field of each record. The schema defines the format of each record and the data types and acceptable ranges of each value. In addition, it defines which fields are mandatory.

Fields whose value depends on context (and are not defined in the schema) will be validated by explicit rules to make sure that all requirements for their processing are followed.

Order events will be checked for both internal consistencies and valid relationships when referencing other orders or events from the same reporter.

The full lifecycle will be generated from the full set of order events, and any order that is not fully linked or terminated (executed, canceled, or flagged as routed to a foreign destination) will be flagged as an error.

Receipts will be generated for each phase. The feedback files will have the following extensions for each stage:

- INGESTION - .ingestion
- LINKAGE_DISCOVERY - .linkage

The feedback files will contain the following fields:

- CAT Submitter ID
- CAT Reporter ID
- Timestamp of Receipt
- Business Date
- Stage - INGESTION or LINKAGE_DISCOVERY
- Status - Success if all records in the file were accepted, Failure if some or all records were rejected, or Warning if some or all records have warnings
- Number of Accepted Records - this stage only
- Number of Rejected Records - this stage only
- Failure Filename - if present, contains the relative name of the file in the downloads/failures directory containing records that were rejected.
- Failures - a list of Failures, each containing:
  - Severity - WARNING or ERROR
  - Failure Code
  - Failure Description
- Original Record Index - if applicable, the 0-based index of the record in the submitted file
- Failure Record Index - if applicable, the 0-based index of the record in the Failure Report

The following is an example JSON object for a successful OrderEvents ingestion:

```json
{
    "submitter": "SUBID",
    "reporter": "MYID",
    "receiptTimestamp": "20170307T153552.000001089",
    "businessDate": 20170307,
    "stage": "INGESTION",
    "status": "Success",
    "acceptedCount": 214513134,
    "rejectedCount": 0
}
```

The following is an example JSON object for an unsuccessful OrderEvents ingestion:

```json
{
    "submitter": "SUBID",
    "reporter": "MYID",
    "receiptTimestamp": "20170307T153552.000001089",
    "businessDate": 20170307,
    "stage": "INGESTION",
    "status": "Failure",
    "acceptedCount": 214513132,
    "rejectedCount": 2,
    "failureFile": "MYID_20170307_OrderEvents_000002.rejects.pgp",
    "failures": [
        {
            "severity": "ERROR",
            "code": "OE.INGEST.014",
            "desc": "Invalid value for TIF",
            "origIdx": 123456,
            "failIdx": 0
        },
        {
            "severity": "WARNING",
            "code": "OE.INGEST.009",
            "desc": "Symbol has been delisted",
            "origIdx": 654321,
        }
    ]
}
```
12.5. Corrections

Corrections may be made manually via the web portal or uploading correction data files. If a failure is reported as individual errors (less than 10 percent of the file) or the entire file was rejected, then corrections should be submitted as repair records.

Corrections may be made at any time, even if beyond the error correction timeframe.

12.5.1. Repair Record

Repair records may be submitted to correct or delete a previously submitted record. Both records that have been previously rejected and records that have already been accepted may be repaired.

Repair records should be submitted in a repair file. The repair filename must be the filename of the original repair file, with the string "Corrections" appended to the original file kind. For example, if the original filename was MYID_20170101_OrderEvents_000123.[json|csv], then the repair filename should be MYID_20170101_OrderEventsCorrections_000001.[json|csv]. The metadata file should be similarly named, with an extension of .meta.

Metadata for the repair record should be valid for the repair record file, not the original file being repaired. For example, the Encrypted Hash and Encrypted Size should be the hash and size of the repair record file. The Business Date, CAT Submitter ID, and CAT Reporter ID should be the same for the repair record file as the original file being repaired.

Each repair record will uniquely identify a record to repair using the 0-based index of the original record. If the record to repair cannot be identified, the repair record will be rejected. If a feedback file has a failure file associated with it, then the index will be pre-populated on Correction Records in the failure file to ease repair submissions. Otherwise, if correcting or deleting non-rejected records, the submitter will need to populate these fields itself.
Failure files (as indicated by the `failureFile` field on some feedback JSON objects) can be used to submit corrections. Since the files are pre-populated with Correction Records, a file only needs to be downloaded, the appropriate changes to the contents of the record(s) made, and the file submitted with the appropriate file name.

If the number of rejected records exceeds a predefined threshold (10 percent of the number of records in the original file), the entire repair record file will be rejected.

Repair record files may contain delete and correction records only. If an invalid record type is detected in the file, then the entire file will be rejected.

### 12.5.1.1. Delete Record

A delete record must contain the following fields:
- Record Type - DEL
- Original Record Index - the 0-based index of the record in the original file

The following is an example delete record:

```json
{
    "type": "DEL",
    "recordIdx": 456
}
```

### 12.5.1.2. Correction Record

A correction record must contain the following fields:
- Record Type - COR
- Original Record Index - the 0-based index of the record in the original file
- Replacement record - the corrected record

The following is an example correction record:

```json
{
    "type": "COR",
    "recordIdx": 456,
    "newRecord": {
        "type": "MENO",
        "reporterIMID": "FRMA",
        "eventTimestamp": "20170801T143031.123456",
        "symbol": "XYZ",
        "orderID": "012345",
    }
}
```
12.5.1.3. Repair Record Feedback

Repair record files will receive the same types of feedback as the file being repaired.
13. Testing

CAT will provide an environment for testing that mirrors the current functionality of the CAT production environment, as well as including functionality for next release version of the CAT environment when available. The CAT testing environment will automatically determine which specification version reporters are using for submissions. If Error reporting formats change, reporters will receive feedback in the current and new specification via ftp, as well as have access to current/new web portal urls for specification changes that impact the web portal. Current/new connectivity changes will also be supported concurrently.

The testing environment does perform lifecycle linkage, and reporters are encouraged to coordinate testing with their counterparties so as to test lifecycle linkage with their counterparties. Without simultaneous contra-party reporting in the test environment, firms will not be able to test linkage with their counterparties.

Firms should test their submissions using the testing environment before they begin submitting to the production environment.

Please note that reporters should not submit highly sensitive data, such as PII data, into the testing environment.

The test environment is available 24 hours a day, 6 days a week. Refer to the CAT website for contact information and hours of operation for support.

Firms connect to the test environment in the same manner they would connect to the production environment, however, for the connection to the test environment, one or more alternate IP/domains may be used.

Testing does not relieve a firm of its responsibilities to submit production data to the CAT system.
14. CAT Help Desk and Support

- **Helpdesk** - The CAT Help Desk is currently available to Industry Members. For questions and support please call or email at helpdesk@thesyscat.com

- **Web announcements** - Web announcements will be made available on the public website (www.catnmsplan.com). You can also subscribe to receive email notifications regarding changes to the website. These announcements are used to post information related to the operation of CAT. The Public Website can be accessed via Internet, and is sourced outside the CAT secure network.

- **Reporting Statistics** - reporting statistics will be made available via the CAT Reporter Web Portal on a daily basis, and are posted when processing for all files has completed. The daily statistics will include, at a minimum, the following information for order events and reference data:
  - CAT Reporter ID;
  - Date of Submission;
  - Number of files received;
  - Number of files accepted;
  - Number of files rejected;
  - Number of total order events received;
  - Number of order events accepted;
  - Number of order events rejected;
  - Number of each type of report received;
  - Number of each type of report accepted;
  - Number of each type of report rejected;
  - Number of unknown accounts;
  - Number of late submissions;
  - Order-IDs rejected;
  - Reasons(s) for rejection;
  - Number of records attempted to be matched;
  - Number of records matched; and
  - Percentage of records matched.

- **System Status** - CAT system status (e.g., outage, maintenance, other notifications, etc.) can be viewed on the Reporter Web Portal.
15. Appendix B

15.1. Failure Codes

A failure code is a machine-parseable description of why a file or record was rejected. This differs from a failure description, which is intended for human consumption. Each failure code is divided into a failure category, sub-category, and value, joined together by a period. Each category roughly corresponds to the stage of processing at which a file or record was rejected. The following failure categories are defined:

- **FILE** - a problem with file name or permissions, with the following sub-categories:
  - NAME - a problem with the file name
  - PERM - a problem with file permissions
  - TIMEOUT - a timeout waiting for the corresponding data or meta file

- **INT** - a problem with file metadata or hash, with the following sub-categories:
  - META - an incorrect or mismatched metadata value

- **OE** - a problem with an order event file or record, with the following sub-categories:
  - COUNT - fewer or more records in the file than specified by the Record Count
  - INGEST - a problem with an individual record encountered in the ingestion stage
  - LINK - a problem with an individual record encountered in the linkage discovery stage

A failure code may be used for warnings or errors, distinguished by the severity field in the failure report. The failure code itself does not distinguish between a warning (in which a record is accepted but still shows up in the failure report) and an error (which causes a record to be rejected).