September 15, 2021
Yoshimasa (Yoshi) Satoh, CFA

[0.1] Introduction

Nasdaq Trade Surveillance (**NTS**, formerly known as SMARTS) provides a **direct data access to your company's triggered alerts via Alerts XML**. NTS users need the Alerts/Alerts Management permission for **Alerts XML**. It is provided to NTS users as part of NTS products and services, and can be used without any extra cost.

NTS users can download historical triggered alerts as **a single XML file** and **CSV files** on a daily basis via **Alerts XML**. The former XML file includes all the alerts generated for a specific day; the latter CSV files correspond to individual triggered alerts, i.e., one CSV file for one fired alert.

This document describes how NTS users can **[A] download alert data by using Alerts XML** and **[B] process the alert data (XML files) with the Python** programming language.

With that, your company should be able to execute customized and deeper analysis and gain insight on your end. Statistical/machine learning techniques could be applied as well.

Please be advised that this document and related services are provided only for our NTS users' feedback and Nasdaq's evaluation purposes; it is not part of officially provided products and services by Nasdaq, and Nasdaq may cease this client feedback and evaluation process without notice. Please let the author of this document (Yoshimasa Satoh, CFA Yoshimasa.Satoh@nasdaq.com) know if you have any comment, question, or request. Your feedback would be extremely important for our evaluation.

[0.2] References

For more detail of **Alerts XML**, please refer to documents on the Knowledge Base by Nasdaq:
https://customer-support.nasdaq.com/kb/display/NTSKB/Technical+Specifications

- **Alert XML export specification** (NTS-XML-Interface-specificationV1.14-Final.pdf)

[1] Using Alerts XML to download a zip file

You can use the following URL to download bundled alerts:
https://**<yourorganisation>**.smartsbroker.com/cmss/citadel/exportAlerts?**marketCode**=**<market>**&**date**=**<yyyymmdd>**&**bundle**=**<true|false>**&**apiVersion**=**<n>**&**lookbackDays**=**<n>**

where:

- **marketCode** = market name (e.g., **asx**)
- **date** = format "yyyymmdd" specifying an end date of alerts to be exported. (e.g., **20210730**)
- **bundle** = This takes a "true" or "false" value specifying whether to export alerts as a zip archived file (e.g., **true**)
- **apiVersion** = This takes a number between 5 and 9 specifying the API version of the export format. See for more information about each version of export format. Please use 5 or higher as we do not support lower API versions. (e.g., **8**)

- **lookbackDays** = This specifies how many calendar days (not business/trading days) from the specified "end date" are looked back for exporting. For example, "lookbackDays=6" with no specified date will export the current day's alerts plus 6 calendar days of alerts, i.e., 7 calendar days in total. For example, if you use **lookbackDays=6** and **date** = **20210730**, which is Friday, then 7 calendar days from 24th (Saturday) to 30th (Friday) July 2021 will be covered. This lookbackDays parameter is usually limited to 7 days on purpose to mitigate possible server performance issues. If you would like to export 7-day / weekly data on a certain day, then please use "lookbackDays=6" as it always covers 7 calendar days. If you need to remove this limit temporarily, please contact our client support team (Market Technology Service Desk MTSD@nasdaq.com) to help. (e.g., **6**)

As mentioned above, the URL can go like this:
https://**<yourorganisation>**.smartsbroker.com/cmss/citadel/exportAlerts?**marketCode=asx**&**date=20210730**&**bundle=true**&**apiVersion=8**&**lookbackDays=6**

[2] Unzipping a downloaded file

If you download an AlertsExport.zip file, unzip the file, and then you'll see a directory structure as follows:

- **AlertsExport**
  - **summary.xml**
  - (directory with a market name, e.g., **asx**)
    - (directory with yyyymmdd, e.g., **20210730**)
      - **alerts.xml**
      - (CSV files for each alert triggered)

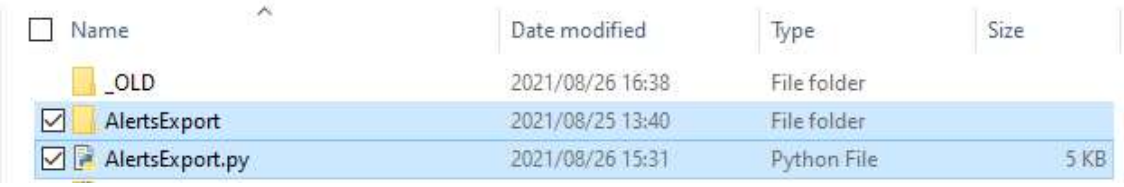
summary.xml shows summary of data, such as date when alerts were generated, alertCount for the number of alerts for each day, etc.

alerts.xml for each day describes the detail of all the triggered alerts.

Please note that there will be no yyyymmdd directory for a day without alert, which is most likely a non-business/trading day.

[3] Preparing a Python script

Please access the following URL on Github,
https://github.com/yoshisatoh/nasdaq/tree/main/NTS/Alerts_XML/AlertsExport.py
and then download the **AlertsExport.py** file from the "Raw" button and save it to your directory where you extracted the **AlertsExport** directory as described above. Namely, the **AlertsExport.py** file is expected to be saved on the directory where the **AlertsExport** directory is located.



[4] Executing a Python script

After setting up **Python** on your computer, please execute the **AlertsExport.py** script on your Command Prompt (Windows) or Terminal (MacOS) as follows:

python AlertsExport.py

[5] Reviewing results

After the execution above, the following files will be created; sample results are also shown:

summary.markets.market.csv (A certain market specified in the URL)

```
asx
```

summary.total.csv (The number of total alerts in a specified period for a certain market)

```
500
```

summary.markets.csv (The number of total alerts per calendar day for a certain market)

```
asx,20210724,0
asx,20210725,0
asx,20210726,100
asx,20210727,110
asx,20210728,120
asx,20210729,70
asx,20210730,100
```

(market).(yyyymmdd).alerts.csv

```
alert_id,market,date,alertCount,time,title,security,house,trader,account_ref
asx-20210726-1,asx,2021-07-26,1,09:30:00.000,POSSIBLE INSIDER TRADING (HOUSE) (BUY),XXX,000,NA,NA
asx-20210726-2,asx,2021-07-26,2,10:00:00.000,OPENING PRICE JUMP (HOUSE),XXX,000,TRD1,ACT1
…
asx-20210726-99,asx,2021-07-27,99,16:00:00.000,POSSIBLE INSIDER TRADING (HOUSE) (SELL),
```

```
XXX,000,TRD1,ACT1
```

(market).png

This is a bar chart image file corresponding to **summary.total.csv** and **summary.markets.csv**.

(market).all.alerts.csv

```
alert_id,market,date,alertCount,time,title,security,house,trader,account_ref
asx-20210726-1,asx,2021-07-26,1,09:30:00.000,POSSIBLE INSIDER TRADING (HOUSE) (BUY),XXX,000,,
asx-20210726-2,asx,2021-07-26,2,10:00:00.000,OPENING PRICE JUMP (HOUSE),XXX,000,TRD1,ACT1
…
asx-20210730-99,asx,2021-07-30,99,16:00:00.000, SIGNIFICANT VOLUME IN LAST 2 MINUTES OF TRADING (ACCOUNT) (BUY) (WITH PRICE IMPACT),XXX,000,TRD1,ACT1
```

(market).all.alerts.counts.csv

```
,alerts
TRADE TO TRADE (BUY) (HOUSE),70
TRADE TO TRADE (SELL) (HOUSE),50
…
PATTERN ALERT (OPENING PRICE JUMP (HOUSE)),1
```

(market).all.alerts.counts.house.csv

(market).all.alerts.counts.house.trader.csv

(market).all.alerts.counts.house.account_ref.csv

(market).all.alerts.counts.house.security.csv

(market).all.alerts.counts.house.trader.account_ref.security.csv

These files contain aggregated alert counts for each set of columns.

(market).all.alerts.marketCode.securityCode.type.oid.csv

```
alert_id,marketCode,securityCode,type,oid
asx-20210726-2,asx,XXX,trade,1234567890
…
```

[6] Contacting the author

Please let the author of this document (Yoshimasa Satoh, CFA Yoshimasa.Satoh@nasdaq.com) know if you have any comment, question, or request. Your feedback would be extremely important for our evaluation. The author will make an effort to update **AlertsExport.py** based on your requests.