

## [0.1] Introduction

**Nasdaq Trade Surveillance (NTS, formerly known as SMARTS Trade Surveillance)** provides an **alert data direct access**. It is called **Alerts XML**, and provided to NTS users as part of standard NTS products and services; namely, it does not incur any extra cost. An NTS user needs an Alerts/Alerts Management permission for Alerts XML, which can be granted by your company's NTS User Manager.

NTS users can download historical data of triggered alerts as **a single XML file (alerts.xml)** and **CSV files**. 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 alert.

This document describes how NTS users can **[A] download alert data by using Alerts XML** and **[B] process the alert data** (one alerts.xml file for one day) **with the Python programming language**. With that, your company should be able to execute customized and deeper data-centric analysis and gain insights on your end. For instance, statistical/machine learning techniques could be applied to the data.

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](mailto:Yoshimasa.Satoh@nasdaq.com)) know if you have any comment, question, or request. Your feedback would be extremely important for our evaluation.

## [0.2] Reference(s)

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?apiVersion=<n>&bundle=<true|false>&marketCode=<market>&date=<yyyymmdd>&lookbackDays=<n>>

where:

- **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**)
- **bundle** = This takes a "true" or "false" value specifying whether to export alerts as a zip archived file (e.g., **true**)
- **marketCode** = market name (e.g., **asx**)
- **date** = format "yyyymmdd" specifying an end date of alerts to be exported. (e.g., **20211029**)

- **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. Moreover, if you use **lookbackDays=6** and **date = 20211029**, which is Friday, then 7 calendar days from 23rd (Saturday) to 29th (Friday) October 2021 will be covered. The maximum number of lookbackDays is usually set to 1 as a default setting; if you need to remove this limit temporarily, please contact our client support team (Market Technology Service Desk [MTSD@nasdaq.com](mailto:MTSD@nasdaq.com)) to help. (e.g., **if you omit this, then alerts on a single specific date will be downloaded.**)

As mentioned above, a URL can go like this:

<https://<yourorganisation>.smartsbroker.com/cmss/citadel/exportAlerts?marketCode=asx&date=20211029&bundle=true&apiVersion=8>

Similarly, you can repeat this four times for other dates, such as, **date=20211028, 20211027, 20211026, and 20211025**. In this document, these five zip files will be used.

## [2] Unzipping a downloaded file

If you download an **AlertsExport.zip** file for the date **20211025**, unzip the file, and then you’ll see a directory structure as follows:

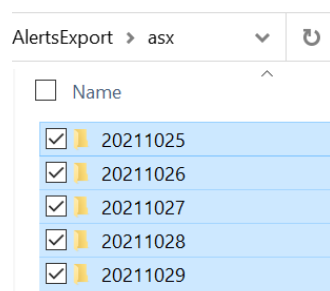
- **AlertsExport**
  - **summary.xml** (See **Appendix**)
  - (directory with a market name, e.g., **asx**, **potentially cross markets like australia and/or australiad are created as well**)
    - ✧ Directory **yyyymmdd**, e.g., **20211025**.
      - **alerts.xml**
      - (CSV files for each alert 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.

Similarly, if you repeat this for other dates, i.e., **20211026, 20211027, 20211028, and 20211029**, then you can find directories **yyyymmdd** under **asx** directories; please copy and paste **yyyymmdd** directories so that you can have all the directories here. Please note that there will be no **yyyymmdd** directory for a day without alert, which is most likely a non-business/trading day **yyyymmdd**.

Last Update: November 5, 2021  
Yoshimasa (Yoshi) Satoh, CFA






### [3] Preparing a Python script

Please access the following URL on Github,

[https://github.com/yoshisatoh/nasdaq/tree/main/NTS/Alerts\\_XML/AlertsExport.py](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 created and updated 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.

<input type="checkbox"/> Name	Date modified	Type	Size
 _OLD	2021/08/26 16:38	File folder	
<input checked="" type="checkbox"/>  AlertsExport	2021/08/25 13:40	File folder	
<input checked="" type="checkbox"/>  AlertsExport.py	2021/08/26 15:31	Python File	5 KB

### [4] Executing a Python script

After setting up **Python (3.9.7 or later)** and Python libraries used in **AlertsExport.py** 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** (A number of total alerts in a specified period in a certain market)

500

**summary.markets.csv** (A total alert count per calendar day in a certain market)

```
asx,20211023,0
asx,20211024,0
asx,20211025,100
asx,20211026,110
asx,20211027,120
asx,20211028,70
asx,20211029,100
```

**(market).png**

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

**(market).(yyyymmdd).alerts.csv** (Alerts for a calendar day in a certain market)

```
alert_id,market,date,alertCount,time,title,security,house,trader,account_ref
asx-20211025-1,asx,2021-10-25,1,09:30:00.000,POSSIBLE INSIDER TRADING (HOUSE)
(BUY),XXX,000,NA,NA
asx-20211025-2,asx,2021-10-25,2,10:00:00.000,OPENING PRICE JUMP (HOUSE),XXX,000,TRD1,ACT1
...
asx-20211025-99,asx,2021-10-25,99,16:00:00.000,POSSIBLE INSIDER TRADING (HOUSE) (SELL),
XXX,000,TRD1,ACT1
```

**(market).all.alerts.csv** (All alerts in a specified time period in a certain market)

```
alert_id,market,date,alertCount,time,title,security,house,trader,account_ref
asx-20211025-1,asx,2021-10-25,1,09:30:00.000,POSSIBLE INSIDER TRADING (HOUSE) (BUY),XXX,000,,
asx-20211025-2,asx,2021-10-25,2,10:00:00.000,OPENING PRICE JUMP (HOUSE),XXX,000,TRD1,ACT1
...
asx-20211029-99,asx,2021-10-29,99,16:00:00.000, SIGNIFICANT VOLUME IN LAST 2 MINUTES OF
TRADING (ACCOUNT) (BUY) (WITH PRICE IMPACT),XXX,000,TRD1,ACT1
```

**(market).all.alerts.counts.csv** (Total alert counts per alert in a specified time period in a certain market)

```
,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**

Last Update: November 5, 2021  
Yoshimasa (Yoshi) Satoh, CFA

**(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 total alert counts for each set of columns.

**(market).all.alerts.marketCode.securityCode.type.oid.csv** (Alerts with trade/order type and order ID)

alert_id,marketCode,securityCode,type,oid
asx-20211025-2,asx,XXX,trade,1234567890
...

## [6] Contacting the author

Please let the author of this document (Yoshimasa Satoh, CFA [Yoshimasa.Satoh@nasdaq.com](mailto: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.

### Appendix: summary.xml

When you download your daily files without the **lookbackDays** option, a **summary.xml** file for the first date has to be updated as follows:

#### summary.xml

```
...
<summary>
  <total>
    <alertCount>500</alertCount>
    <attachmentCount>450</attachmentCount>
  </total>
  <markets>
    <market marketCode="asx">
      <day date="20211025">
        <alertCount>100</alertCount>
        <attachmentCount>90</attachmentCount>
      </day>
      <day date="20211026">
        <alertCount>110</alertCount>
        <attachmentCount>100</attachmentCount>
      </day>
      <day date="20211027">
        <alertCount>120</alertCount>
        <attachmentCount>110</attachmentCount>
      </day>
      <day date="20211028">
        <alertCount>70</alertCount>
        <attachmentCount>60</attachmentCount>
      </day>
      <day date="20211029">
        <alertCount>100</alertCount>
        <attachmentCount>90</attachmentCount>
      </day>
    </market>
  </markets>
</summary>
```

<alertCount>**500**</alertCount> can be calculated by summing alertCount for the entire period:

$$100 + 110 + 120 + 70 + 100$$

Similarly, <attachmentCount>**450**</attachmentCount> is derived by 90+100+110+60+90.