

**OCEAN
NETWORKS
CANADA**

OCEANS 2.0 API PROGRAMMATIC ACCESS TO ONC DATA UPDATE

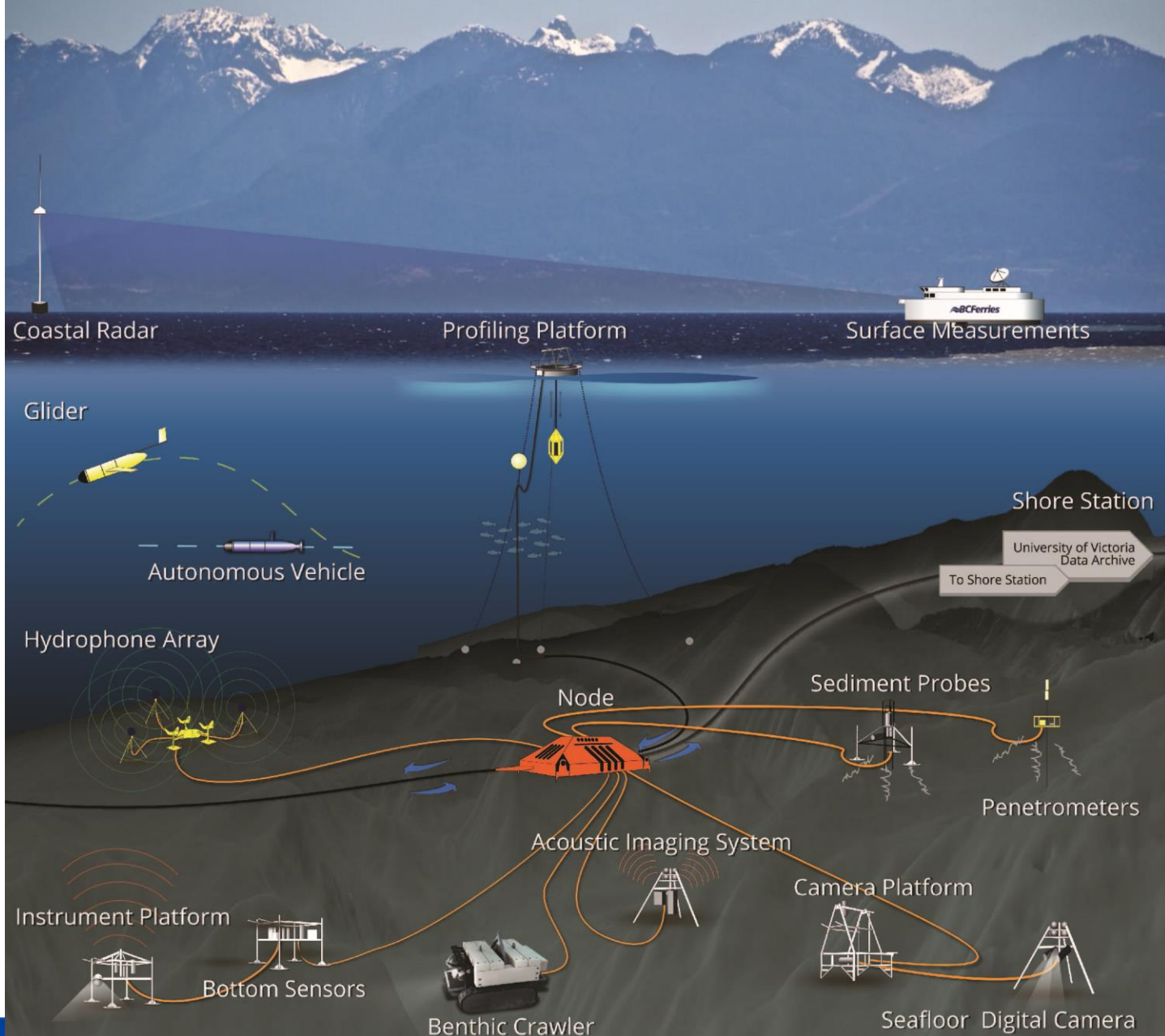
Allan Rempel | 26 February 2018

OCEAN NETWORKS CANADA

Networks of cabled observatories feeding a data archive



OBSERVATORY INFRASTRUCTURE



OCEAN NETWORKS CANADA

Networks of cabled observatories feeding a data archive

[Ocean Networks Canada: By the Numbers](#)

- **2** regional and 4 community observatories
- **7** shore stations
- **850+** km seafloor backbone cables
- **over 50** instrumented sites with platform
- **7** mobile instrument platforms
- **400** instruments containing **over 5000** sensors online 24/7/365
- **2006** – the year data began to flow from the VENUS observatory in Saanich Inlet
- **500+** terabytes of data archived in over **26 million** files
- **280** gigabytes of data collected every day
- **35** gigabytes of data are distributed every day
- **\$0.00** – your cost to use the data



ONC API

Programmatic access to ONC's Data Archive

The Empower Project

The Web Services

- Discovery
- Data Delivery

The Client Libraries

- Python
- MATLAB
- R *

The Documentation

- Samples
- Client libraries
- Use Cases

Demos

Testing Plan

Road Ahead

ONC API

Empower Project Overview

- A Research Platform for User-Defined Oceanographic Data Products
- 2 Year project funded by CANARIE
- Two Phases

API

Provide researchers with programmatic access to the Oceans 2.0 data products through a specially designed Application Programming Interface (API)

Sandbox

Enable researchers to define, test, use and share processing code for user-defined data products in a custom-designed programming environment.

ONC API

API Phase Overview

User-Centred design approach

- 1) Identify existing and future users
- 2) Develop hypothesis based on domain knowledge and current usage
- 3) Develop questionnaires to validate hypothesis and elicit responses about:
 - a) Research focus
 - b) Data needs
 - c) Analysis needs
 - d) Analytical tools
 - e) Usage and performance expectations
 - f) Publishing needs
- 4) Perform interviews and capture responses
- 5) Analyze responses and organize into functional groups or clusters of requirements
- 6) Develop personas to characterize user goals and behaviours
- 7) Develop use cases to capture the functional needs and requirements of the personas
- 8) Develop user stories to inform the functional design
- 9) Build the API in an iterative cycle of sprints with user feedback and testing throughout

ONC API

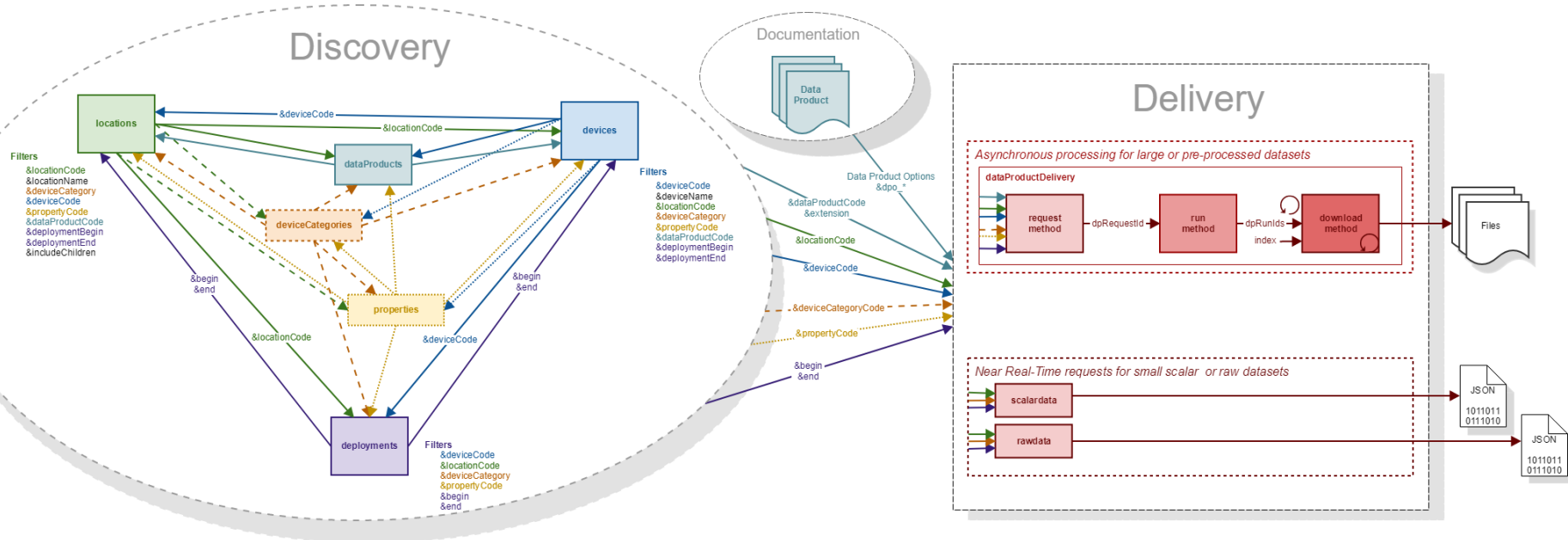
API Phase Overview

Results

- 12 user interviews and questionnaires
- 14 existing and future users
- [4 Use Cases](#)
 - Bird Studies Canada
 - Ouranos
 - Academic
 - ONC Internal
- 3 new personas
 - Lead Researcher
 - Scientific Modeller
 - Scientific Developer
- 54 User Stories
- 4 functional groups
 - Discovery
 - Filtering
 - Scripting
 - Internal Needs

ONC API

Understanding the web services



- Data Discovery
 - 6 web services to help discover what data can be downloaded
 - [Where is data available](#)
 - [What devices are available](#)
 - [What device categories are available](#)
 - [What properties are available](#)
 - [What data products are available](#)
 - [When are devices deployed](#)

- Data Delivery (Download)
 - [1 web service to download data as customizable data product](#)
 - Request a Data Product be created
 - Download the Data Product once complete
 - 2 web services for near real-time data access
 - [Scalar data](#)
 - [Raw data](#)

ONC API

Data Discovery Services

6 Service Endpoints

[.../api/locations](#)

- ONC search tree nodes (aka, stations) that data can be downloaded from
- Uniquely identified by locationCode.
e.g. BACME (*Barkley Canyon / MidEast*)

[.../api/devices](#)

- Instruments that have one or more sensors that observe a property or phenomenon with a goal of producing an estimate of the value of a property
- Uniquely identified by deviceCode
e.g. CAM-TEMPO-MINI-2 (*Tempo-Mini Colour Video AXIS Q1755 [Camera 2]*)

[.../api/deviceCategories](#)

- Device Category grouping
- Uniquely identified by deviceCategoryCode
e.g. VIDEOCAM (*Video Camera*)
DSC (*Still Camera*)

[.../api/properties](#)

- Observable phenomenon (aka, variables)
- Common name given to sensor types (e.g. oxygen, pressure, temperature, etc.)
- Uniquely identified by propertyCode

[.../api/dataProducts](#)

- Data Products available for download
- Uniquely identified by dataProductCode & extension
e.g. 3DCIS (*3D Camera Image Stitching*) & tar
MP4V (*MP4 Video*) & mp4

[.../api/deployments](#)

- Instrument deployments
- Uniquely identified by deviceCode & dates
- Each deployment has a location code, device code, lat/long/depth, heading/pitch/roll, has-data flag, and deployment date range.

ONC API

Data Discovery Services

- Reductive Filtering
 - Additional filters can be added to the url to reduce the selection

Example - return all locations that have a Video Camera deployed between two dates

```
http://data.oceannetworks.ca/api/locations?method=get
&token=x6ecb030-1835-1dc5-29nc-q58l7wjpl607
&deviceCategory=VIDEOCAM
&deploymentBegin=2016-07-27T00:00:00.000Z
&deploymentEnd=2016-08-01T23:59:59.999Z
```

- Returned as a JSON payload

```
[{"dataProductCode":"AD",
  "dataProductName":"Audio Data",
  "extension":"wav",
  "helpDocument":"https://wiki.oceannetworks.ca/display/DP/7"}]
```

ONC API

Data Product Delivery

- Three step process using dataProductDelivery web service

1. Request a data product using the **request** method

- Include data product filters in URL
 - locationCode and deviceCategoryCode *or* locationCode, deviceCategoryCode and propertyCode *or* deviceCode *or* deviceCode and propertyCode
 - dataProductCode and extension
 - begin and end
 - Data Product Options vary with data product
- Returns a RequestId
- Returns run time and size estimates
 - Can be used for request verification

```
https://data.oceannetworks.ca/api/dataProductDelivery?method=request&locationCode=BACAX&deviceCategoryCode=ADCP2MHZ&dataProductCode=LF&extension=txt&begin=2016-07-25T00:00:00.000Z&end=2016-07-29T00:00:00.000Z&token=YOUR\_TOKEN\_HERE
```

```
{"compressedFileSize":25142845,"downloadTimes":{"10Mbps":13.343616,"50Mbps":2.668723,"150Mbps":0.8895744},"dpRequestId":"2615408","fileSize":133436160,"numFiles":4}
```

2. Run the data product using the **run** method

- Include RequestId in URL
- Starts the data product generation process by adding it to the Task Queue

```
https://data.oceannetworks.ca/api/dataProductDelivery?method=run&dpRequestId=2615408&token=YOUR\_TOKEN\_HERE
```

```
[ { "dpRunId": 5991552, "fileCount": 0, "status": "data product running" } ]
```

3. Download the data product using the **download** method

- Include RunId and index in URL
- Informs on process status with messages in the payload HTTP status codes
- Downloads file when process is complete
- Requires same token as run request

```
https://data.oceannetworks.ca/api/dataProductDelivery?method=download&dpRunId=5991552&token=YOUR\_TOKEN\_HERE
```

```
*** no output, you just get the file(s)! ***
```

ONC API

Using the web services

- Via browser
 - Make a simple http request using a browser link to return information
- Via code
 - Any language that supports HTTP requests including:
 - Python *
 - MatLab *
 - R *
 - JavaScript
 - C++
 - Java
 - *Available ONC client libraries
- All requests require a user token
 - Create user at <https://data.oceannetworks.ca/login>
 - Generate token on 'Web Services API' at <http://data.oceannetworks.ca/Profile>
 - Please use your own token
 - It allows us to better understand your data needs and inform you when changes or improvements are made to ONC web services

ONC API

Client Libraries

Client libraries provide

- Quick, easy and consistent access to ONC data and resources
- In scientific programming language of choice
- With minimal lines of code

Currently available

-  python 2.7+
- Matlab  R2017a
-  3.3+

<https://wiki.oceannetworks.ca/display/O2A/Client+Libraries>

ONC API

Using the Client Libraries

- Python

- Install package using pip
- Add to library to script using

```
from onc.onc import ONC
```
- Create ONC object using

```
onc = ONC("YOUR_TOKEN")  
or  
onc = ONC("YOUR_TOKEN", True, False, "c:/ONC/Data")
```

- MATLAB

- Download Add-On Toolbox from ONC Wiki and install
- Create ONC object using

```
o = ONC("YOUR_TOKEN")  
or  
o = ONC("YOUR_TOKEN", true, false, "c:/ONC/Data")
```

- R

- Download package from ONC Wiki and install
- Add library to script using

```
library(onc)
```
- Create ONC object using

```
onc = new("onc", token="YOUR_TOKEN")  
or  
onc = new("onc", token="YOUR_TOKEN", production=TRUE, showInfo=FALSE, outPath="c:/ONC/Data")
```

ONC API

Client Libraries – Single line of code

Download Time Series Scalar Data Product in CSV format for ADCP 2 MHZ at Barkley Canyon - Axis

- Python

```
results = onc.orderDataProduct({'locationCode': 'BACAX',  
                                'deviceCategoryCode': 'ADCP2MHZ',  
                                'dataProductCode': 'TSSD',  
                                'extension': 'csv',  
                                'begin': '2016-07-27T00:00:00.000Z',  
                                'end': '2016-08-01T00:00:00.000Z',  
                                'dpo_qualityControl': 1, 'dpo_resample': 'none', 'dpo_dataGaps': 0})
```

- MATLAB

```
results = o.orderDataProduct(struct('locationCode', 'BACAX', ...  
                                   'deviceCategoryCode', 'ADCP2MHZ', ...  
                                   'dataProductCode', 'TSSD', ...  
                                   'extension', 'csv', ...  
                                   'begin', '2016-07-27T00:00:00.000Z', ...  
                                   'end', '2016-08-01T00:00:00.000Z', ...  
                                   'dpo_qualityControl', 1, 'dpo_resample', 'none', 'dpo_dataGaps', 0));
```

- R

```
results = onc.orderDataProduct(onc, list(locationCode="BACAX",  
                                         deviceCategoryCode="ADCP2MHZ",  
                                         dataProductCode="TSSD",  
                                         extension="csv",  
                                         begin="2016-07-27T00:00:00.000Z",  
                                         end="2016-08-01T00:00:00.000Z",  
                                         dpo_qualityControl=1, dpo_resample="none", dpo_dataGaps=0))
```

ONC API

Documentation

- [Guide](#)
 - Overview of the API
 - Requesting Data Products using the API
- [API Reference](#)
- [Sample Code](#)
- [Client Libraries](#)
- Use Cases – [Research](#) and [Internal](#)
- [Oceans 2.0 Knowledge Base](#)
 - [Data Products and Metadata](#)
 - [Data Products Catalog](#)
 - [FAQ](#)

<https://wiki.oceannetworks.ca/display/O2A/Oceans+2.0+API+Home>

ONC API

Demo

ONC API

Your mission, should you choose to accept it...

- Obtain images captured on June 20, 2016 on a Sony SuperScorpio camera on an ROV Expedition.

https://data.oceannetworks.ca/api/devices?method=get&token=YOUR_TOKEN_HERE&deviceName=Scorpio

```
{"cvTerm":{"device":[]},"dataRating":[],"deviceCode":"JASON2ROVINSITESUPERSCORPIOCAM","deviceId":23546,"deviceLink":"http://data.oceannetworks.ca/DeviceListing?DevicelD=23546","deviceName":"Insite SuperScorpio (Sony HDR-CX560V) on Jason 2 ROV","hasDeviceData":true}}
```

https://data.oceannetworks.ca/api/deployments?method=get&token=YOUR_TOKEN_HERE&deviceCode=JASON2ROVINSITESUPERSCORPIOCAM

```
[{"begin":"2015-09-01T00:00:00.000Z","depth":14.395,"deviceCode":"JASON2ROVINSITESUPERSCORPIOCAM","end":"2015-09-19T00:00:00.000Z","hasDeviceData":true,"heading":264.800000,"lat":48.427825,"locationCode":"JAS2","lon":-126.174489,"pitch":-8.140000,"roll":-1.670000},{"begin":"2016-06-13T00:00:00.000Z","depth":14.395,"deviceCode":"JASON2ROVINSITESUPERSCORPIOCAM","end":"2016-06-26T00:00:00.000Z","hasDeviceData":true,"heading":264.800000,"lat":48.427825,"locationCode":"JAS2","lon":-126.174489,"pitch":-8.140000,"roll":-1.670000}]
```

https://data.oceannetworks.ca/api/dataProducts?method=get&token=YOUR_TOKEN_HERE&deviceCode=JASON2ROVINSITESUPERSCORPIOCAM

```
{"dataProductCode":"JPGF","dataProductName":"JPG File","extension":"jpg","hasDeviceData":true,"hasPropertyData":false,"helpDocument":"https://wiki.oceannetworks.ca/display/DP/54"}}
```

https://data.oceannetworks.ca/api/dataProductDelivery?method=request&token=YOUR_TOKEN_HERE&deviceCode=JASON2ROVINSITESUPERSCORPIOCAM&dataProductCode=JPGF&extension=jpg&begin=2016-06-20T00:00:00.000Z&end=2016-06-21T00:00:00.000Z

```
{"compressedFileSize":109216495,"downloadTimes":{"10Mbps":10.92165,"50Mbps":2.18433,"150Mbps":0.72810996},"dpRequestId":2690016,"fileSize":109216495,"numFiles":30}
```

- What kinds of files can you obtain from a hydrophone in Barkley Sound?
- What is a deviceCategoryCode "CTD" and what properties does it have?
- How many deployments of a deviceCategoryCode "ADCP2MHZ" have there been, and where are they located?

ONC API

After this session...

- Please continue to try other data access calls.
- Data can also be downloaded here: data.oceannetworks.ca
- Please email me with your written feedback and any issues you run across.
 - agrempel@uvic.ca
- User testing will be continuing through March 2018.
- Your feedback will guide further development both by me and the ONC Software team.

ONC API UPDATE

Timeline of recent events

- Nov 2017 – I joined ONC to try to fill the shoes of Ryan Ross
- Dec 2017 – Oceans 2.0 poster at AGU 2017
- Jan 2018 – released new versions of all 3 client libraries
 - Added getLocationHierarchy() for locations.getTree method
 - Better compatibility for Python V2
 - Brought R functionality in line with Matlab and Python
 - Fixed file counting bug; handle empty parameter lists
 - Updated getDirectRaw*() for new features in rawdata service
- Feb 2018 – released new versions of all 3 client libraries
 - Added user test case scripts to illustrate coverage of user stories
 - Improved error reporting
 - Updated getDirectScalar*() for new features in scalardata service
- Feb 2018 – beta testing with grad students from CS, ECE, SEOS
- Nov-Feb – continuing internal & external testing of API and client libraries
- Mar 2018 – initial version of Sandbox to be released
 - Internal and external user testing
- Apr-Jun – continuing: documentation, testing, fixes, user test case scripts

ONC API

Feedback and discussion...

- Did you have unexplained data unavailability at times?
- Did you find typos or errors in code or documentation?
- Did you think the API was easy enough to use?
- Did you think the client libraries were easy enough to use?
- Did you find the sample code helpful in learning the system?
- Did the overall design (discovery and delivery services) make sense?
- Were the error messages helpful in figuring out and fixing problems?

ONC API UPDATE

Grad student beta testing phase

- Call for beta testers – 13 responded, 10 attended kickoff session
- 2018-02-08 – 2-hour kickoff session: 1-hour presentation + 1-hour lab
 - Some had to leave early but at least 5 stayed the full 2 hours
 - Very engaged, asked good questions, took it seriously
- 2018-02-22 – 1-hour follow-up session: 3 returned to discuss experience
 - Students emailed me written feedback including code
- I'm following up with the other 7...
- Survey – 7-point Likert scale for 10 questions:
 - 1. The data I was trying to access was always available (ie. no data outages).
 - 2. The documentation was fairly clear and error-free.
 - 3. The web services API was fairly straightforward and easy to use.
 - 4. The client libraries were fairly straightforward and easy to use.
 - 5. The sample code was useful in learning how to use the system.
 - 6. The kickoff presentation was useful in learning how to use the system.
 - 7. The overall design of the system (ie. the breakdown into discovery and delivery services) made sense and was a good way to access the huge archive.
 - 8. The error messages returned were reasonably helpful in refining the query.
 - 9. I was generally able to find the data I was looking for.
 - 10. The system was fast and responsive enough for my expectations.
- Preliminary results: only 3 surveys returned so far; 1-3, 3-4, 3-5, 13-6, 10-7
 - Average response: 5.93, between 4 (neutral) and 7 (strongly agree)
 - 2 students used Python, 1 used R, 2 used direct web API calls

OCEAN
NETWORKS
CANADA

THANK YOU!

Ocean Networks Canada is funded by the Canada Foundation for Innovation, Government of Canada, University of Victoria, Government of British Columbia, CANARIE, and IBM Canada.

 @ocean_networks  OceanNetworksCanada visit: oceannetworks.ca