

MATLAB Client Library

- [Requires](#)
- [Installation](#)
- [Versions](#)
- [Classes](#)
 - [ONC Class](#)
 - [Description](#)
 - [Constructor](#)
 - [onc\(\)](#)
 - [Usage](#)
 - [Discovery methods](#)
 - [getLocations\(\)](#)
 - [getLocationHierarchy\(\)](#)
 - [getDevices\(\)](#)
 - [getDeployments\(\)](#)
 - [getDeviceCategories\(\)](#)
 - [getProperties\(\)](#)
 - [getDataProducts\(\)](#)
 - [Data Product Download methods](#)
 - [orderDataProduct\(\)](#)
 - [requestDataProduct\(\)](#)
 - [runDataProduct\(\)](#)
 - [downloadDataProduct\(\)](#)
 - [getDataProductUrls\(\)](#)
 - [downloadFile\(\)](#)
 - [decodeJsonFromUrl\(\)](#)
 - [Near Real-Time data product methods](#)
 - [getDirectScalarByStation\(\)](#) - deprecated
 - [getDirectScalar\(\)](#)
 - [getDirectRawByStation\(\)](#) - deprecated
 - [getDirectRawByLocation\(\)](#)
 - [getDirectRawByDevice\(\)](#)
 - [getDirectFiles\(\)](#)

The ONC MatLab client library contains a number of classes and functions that access ONC data, through the Oceans 2.0 API.

Requires

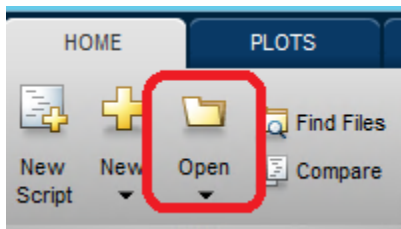


Currently, only **R2017a** is supported, due to updates to the HTTP library that support payloads with non-200 status codes.

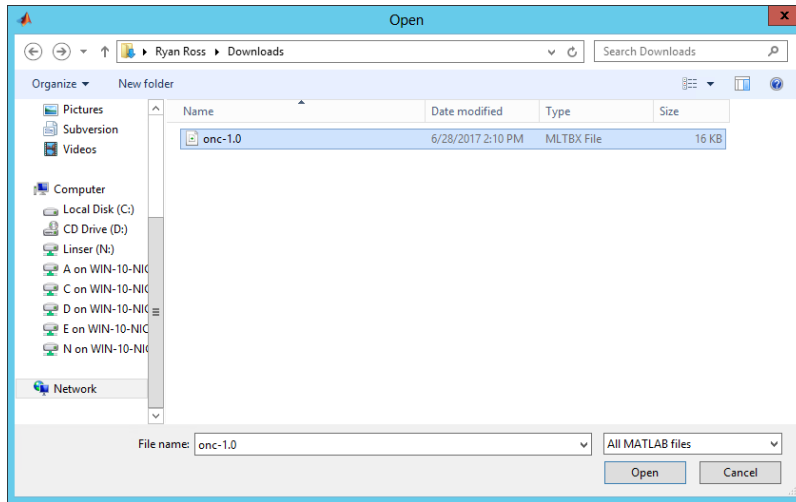
Installation

The ONC client library is an Add-On Toolbox that can be installed in MatLab by performing the following

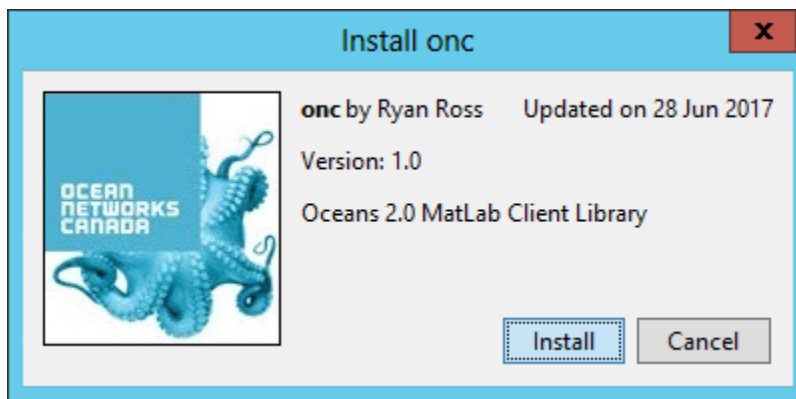
1. Download the latest version of the [ONC Toolbox](#)
2. Open MATLAB **R2017a** and click the 'Open' button.



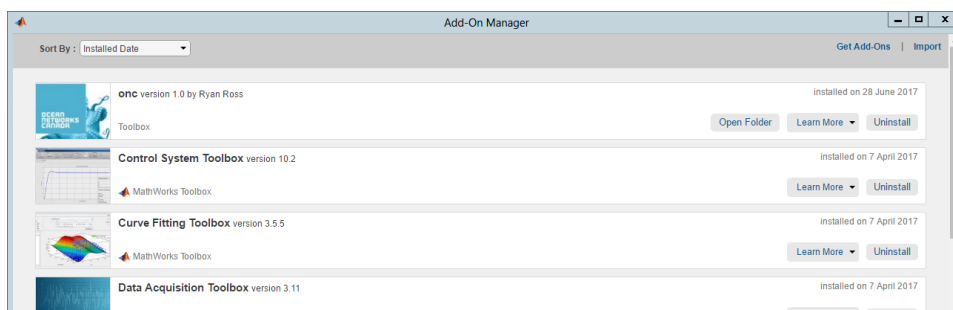
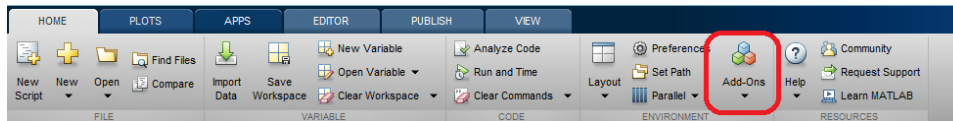
3. Select the downloaded file and click the Open button.



4. When presented with the Install onc dialog, click the Install button.



5. To view the ONC Add-On toolbox click the Add-On button on the Home ribbon and select Manage Add-Ons.



Versions

version	Date	Description	Toolbox	Source
1.0	6/28/2017	Alpha version.	onc-1.0.mltbx	
1.01	7/18/2017	Added <code>getDirectScalar()</code> , <code>getDirectRaw()</code> and <code>getDirectFiles()</code> functions.	onc-1.01.mltbx	onc-1.02-source.zip

1.02	7/28/2017	Resolve unhandled exception with invalid parameters. Resolved naming inconsistencies.	onc-1.02.mltbx	onc-1.02-source.zip
1.03	8/01/2017	Added getDataProductUrls(), downloadFile() and decodeJsonFromUrl() functions	onc-1.03.mltbx	onc-1.03-source.zip
1.04	8/03/2017	Updated production URL from http://dmas.uvic.ca to http://data.oceannetworks.ca .	onc-1.04.mltbx	onc-1.04-source.zip
1.05	8/11/2017	Rename 'parameters' parameter to 'filters' in all applicable onc class functions.	onc-1.05.mltbx	onc-1.04-source.zip
1.06	8/15/2017	Added file parameter default to the download() function.	onc-1.06.mltbx	onc-1.06-source.zip
1.07	8/23/2017	Added getDeployments() function. Updated QA url from http://qaweb2.neptune.uvic.ca to https://qa.oceannetworks.ca . Updated production url to https.	onc-1.07.mltbx	onc-1.07-source.zip
1.08	9/28/2017	Updated orderDataProduct() function to handle estimatedProcessingTime string format Updated downloadDataProduct() function to throttle calls using onc.callsPerSecond when downloadResultsOnly=true	onc-1.08.mltbx	onc-1.08-source.zip
1.09	2018-01-18	Updated Matlab versions of discovery functions to handle empty parameter lists Added getLocationHierarchy() function to access getTree method of locations service Fixed file counting bug in getDirectFiles() function Added getDirectRawBy*() functions for changes in rawdata service Added initial versions of UserTestCase_EMP.UTC_001.m and UserTestCase_EMP.UTC_002.m Updated functions to return more meaningful info when URL call fails	onc-1.09.mltbx	onc-1.09-source.zip
1.10	2018-02-02	Added UserTestCase_EMP.UTC*.m scripts to demonstrate use of client library functions Updated getDirectScalar*() functions for changes in scalardata service Improved error reporting	onc-1.10.mltbx	onc-1.10-source.zip
1.11	2018-04-16	Added UserTestCase_EMP.UTC_004.m to demonstrate use of archivefiles service Updated getDirectFiles() for changes in archivefiles service Improved error handling and reporting in user test case example scripts Updated user test case scripts for change of begin/end to dateFrom/dateTo	onc-1.11.mltbx	onc-1.11-source.zip



To download an attachment, please right click on the link and select **Save Link As, Save Target As** or a similar option provided by your browser. This will open a Save dialog.



New versions are created to resolve known client library issues, add new functionality or handle changes to the underlying Web Services.
The latest version works with the current deployment of the underlying web services. See [API Reference](#).

Classes

class	description
onc	A wrapper class for access to Oceans 2.0 API web services

ONC Class

Description

This class provides a wrapper for some for the most common Oceans 2.0 API requests, including the discovery services (locations, devices, deviceCategories, properties) and delivery (dataProductDelivery, scalardata and rawdata).

Constructor

```
onc(token, production, showInfo, outPath, timeout)
```

Parameter	Type	Description	Example
Required			
token	string	All Web Services require a token. Once logged in at https://data.oceannetworks.ca/login , your token can be retrieved or generated at https://data.oceannetworks.ca/Profile . Click on the "Web Services" tab, then click "Generate Token".	token=YOUR_TOKEN_HERE
Optional			
production	boolean	Indicates whether the ONC Production server URL is used for service requests. <ul style="list-style-type: none">• Default is true.• false is intended for internal ONC testing only.• All non-ONC users must use true or leave blank. Non-ONC user IP addresses are blocked from accessing the Oceans 2.0 QA server.	true or false
showInfo	boolean	Indicates whether verbose script messages, such as request URL and processing time information, are displayed. <ul style="list-style-type: none">• Default is false.• True is intended for script debugging.	true or false
outPath	string	The path that downloaded files are saved to. <ul style="list-style-type: none">• Default is 'c:/Temp'.• Full path will be created if it does not already exist.	'c:/ONC/Download'
timeout	int	The web service request timeout in seconds <ul style="list-style-type: none">• Default is 60	30

Usage

```
o = onc('YOUR_TOKEN_HERE');
```

or

```
production = true;
showInfo = false;
outPath = 'c:/ONC/Data';
timeout = 30;
o = onc('YOUR_TOKEN_HERE', production, showInfo, outPath, timeout);
```

Discovery methods

```
getLocations(filters)
```

Returns a list of locations, as a list of struct objects that meet the criteria defined by the input filters.

Parameter	Type	Description	Example
-----------	------	-------------	---------

Optional			
filters	struct or array	<p>A comma separated list of one or more filter criteria, used to return an exclusive set of location elements from the ONC locations endpoint.</p> <ul style="list-style-type: none"> A list of filter/value pairs in the following format struct('<filter_1>','<value_1>','<filter_2>','<value_2>',...) or {'<filter_1>','<value_1>','<filter_2>','<value_2>',...}. If excluded, all locations are returned. Valid filters: <ul style="list-style-type: none"> locationCode, locationName, deviceCategoryCode, deviceCode, propertyCode, dataProductCode, dateFrom, dateTo, includeChildren See locations Discovery Service for more information on filter usage. 	<pre>struct ('locationCode', 'BACAX') or {'locationCode', 'BACAX'}</pre>

Returns

A list of location objects.

```

    deployments: 50
    locationName: 'Axis '
    depth: 984.3076
    bbox: [1x1 struct]
    description: 'Depth: 985 m Latitude: 48.3167 Longitude: -126.0501 Type: Stationary platform
Description: Canyon axis: benthic processes, biodiversity, sediment dynamics.'
    hasDeviceData: 'true'
    lon: -126.0503
    locationCode: 'BACAX'
    hasPropertyData: 'true'
    lat: 48.3167
    dataSearchURL: 'https://data.oceannetworks.ca/DataSearch?location=BACAX'
```

Property	Type	Description	Example
location	list	A structure array of locations.	
location[].deployments	int	The number of instrument deployments that meet the filter criteria.	deployments : 10
location[].locationName	string	The full name of the location.	locationName: "Axis (POD 1)"
location[].depth	double	The average depth of the deployments.	depth: 75
location[].bbox	struct	A Bounding Box structure.	
location[].bbox.maxDepth	double	The maximum depth of the deployments.	maxDepth: 100
location[].bbox.maxLat	double	The maximum Latitude of bounding box of the deployments.	maxLat: 48.476740
location[].bbox.maxLon	double	The maximum Longitude of the bounding box of the deployments.	maxLon: -123.294904
location[].bbox.minDepth	double	The minimum Depth of the deployments.	minDepth: 50
location[].bbox.minLat	double	The minimum Latitude of bounding box of the deployments.	minLat: 48.47670
location[].bbox.minLon	double	The minimum Longitude of the bounding box of the deployments.	minLon: -123.294900
location[].description	string	The description of the location.	description: "Depth: 985 m Latitude: 48.3167 Longitude: -126.0501 Type: Stationary platform Description: Canyon axis: benthic processes, biodiversity, sediment dynamics."
location[].hasDeviceData	string	Indicates that data products can be requested using a device category for the location.	hasDeviceData: "true"
location[].lon	double	The average longitude of the deployments.	lon: -123.294902
location[].locationCode	string	A code that uniquely identifies a location.	locationCode: "BACAX"

location[].hasPropertyData	string	Indicates that data products can be requested using property code for the location.	hasPropertyData: "true"
location[].lat	double	The average latitude of the deployments.	lat: 48.47672
location[].dataSearchURL	string	The location specific Data Search web page URL.	dataSearchURL: "https://data.oceannetworks.ca/DataSearch?location=BACAX"

Example - Print all locations with 2 MHz ADCP deployed

```
o = onc('YOUR_TOKEN_HERE');

locations = o.getLocations({'deviceCategoryCode', 'ADCP2MHZ'});
for i=1:numel(locations)
    disp(locations(i));
end
```

```
getLocationHierarchy(filters)
```

Returns a hierarchical subtree of locations whose root is defined by the input filter. In this case the only valid input filter is locationCode. In Oceans 2.0 Instruments and Variables are organized by Location so that users can easily drill down by place name or mobile platform name to find the instruments or properties they are interested in.

Parameter	Type	Description	Example
Optional			
filters	struct or array	<p>A comma separated list of one or more filter criteria, used to return an exclusive set of location elements from the ONC locations endpoint.</p> <ul style="list-style-type: none"> • A list of filter/value pairs in the following format struct('<filter_1>','<value_1>','<filter_2>','<value_2>',...) or {'<filter_1>','<value_1>','<filter_2>','<value_2>',...}. • If excluded, all locations are returned. • Valid filters: locationCode • See locations Discovery Service for more information on filter usage. 	<pre>struct ('locationCode', 'BACAX') or {'locationCode', 'BACAX'}</pre>

Returns

A hierarchy of location objects.

```
locationName: 'Axis '
children: null
depth: 984.3076
description: 'Depth: 985 m Latitude: 48.3167 Longitude: -126.0501 Type: Stationary platform
Description: Canyon axis: benthic processes, biodiversity, sediment dynamics.'
hasDeviceData: 'true'
locationCode: 'BACAX'
hasPropertyData: 'true'
```

Property	Type	Description	Example
location[]	list	A list of location structures.	
location[].locationName	string	The full name of the location.	locationName: "Axis (POD 1)"
location[].children[]	list	A list of location structures that are the children of this location structure.	children: null

location[].depth	double	The average depth of the deployments.	depth: 75
location[].description	string	The description of the location.	description: "Depth: 985 m Latitude: 48.3167 Longitude: -126.0501 Type: Stationary platform Description: Canyon axis: benthic processes, biodiversity, sediment dynamics."
location[].hasDeviceData	string	Indicates that data products can be requested using a device category for the location.	hasDeviceData: "true"
location[].locationCode	string	A code that uniquely identifies a location.	locationCode: "BACAX"
location[].hasPropertyData	string	Indicates that data products can be requested using property code for the location.	hasPropertyData: "true"

Example - Print locations at Barkley Upper South Slope

```
o = onc('YOUR_TOKEN_HERE');

locations = o.getLocationHierarchy({'locationCode', 'BACUS'});
if isfield( locations, 'children' )
    for i = 1 : numel( locations.children )
        locations.children(i)
    end
end
```

```
getDevices(filters)
```

Returns a list of devices, as a list of struct objects that meet the criteria defined by the input filters.

Parameter	Type	Description	Example
Optional			
filters	struct or array	<p>A comma separated list of one or more filter criteria, used to return an exclusive set of device elements from the ONC devices endpoint.</p> <ul style="list-style-type: none"> A list of filter/value pairs in the following format struct (<filter_1>,<value_1>,<filter_2>,<value_2>,...) or {<filter_1>,<value_1>,<filter_2>,<value_2>,...}. If excluded, all devices are returned. Valid filters <ul style="list-style-type: none"> deviceCode, deviceName, locationCode, deviceCategoryCode, propertyCode, dataProductCode, dateFrom, dateTo See devices Discovery Service for more information on filter usage. 	<pre>struct ('deviceCode', 'NORTEKADC P9917') or {'deviceCode', 'NORTEKADC P9917'}</pre>

Returns

A list of device objects

```

    cvTerm: [1x1 struct]
        device: [1x1 struct]
            uri: 'http://vocab.nerc.ac.uk/collection/L22/current/TOOL0888/'
            vocabulary: 'SeaVoX Device Catalogue'

    dataRating: [2x1 struct]
        dateFrom: '2008-11-01T00:00:00.000Z'
        dateTo: '2010-05-27T19:27:04.000Z'
        samplePeriod: 1
        sampleSize: 1

        dateFrom: '2010-05-27T19:27:04.000Z'
        dateTo: []
        samplePeriod: 10
        sampleSize: 1

    deviceCode: 'BC_POD1_AD2M'
    deviceId: 11302
    deviceLink: 'https://data.oceannetworks.ca/DeviceListing?DeviceId=11302'
    deviceName: 'Nortek Aquadopp HR-Profiler 2965'

```

Property	Type	Description	Example
device[]	list	A list of devices.	[]
device[].cvTerm	struct	A structure of controlled vocabulary terms.	cvTerm: [1x1 struct]
device[].cvTerm.device[]	struct	A list of structures of controlled vocabulary term information for the device.	device: [1x1 struct]
device[].cvTerm.device[].uri	string	The URI of the controlled vocabulary term.	uri: 'http://vocab.nerc.ac.uk/collection/L22/current/TOOL0888/'
device[].cvTerm.device[].vocabulary	string	The name of the controlled vocabulary to which the term belongs.	vocabulary: 'SeaVoX Device Catalogue'
device[].dataRating[]	list	A list of data ratings structures.	dataRating: [2x1 struct]
device[].dataRating[].dateFrom	datetime string	The starting datetime for a data rating. <ul style="list-style-type: none"> Date Time format: <code>yyyy-MM-dd'T'HH:mm:ss.SSS'Z'</code>. datetime is represented in Coordinated Universal Time (UTC). 	dateFrom: '2008-11-01T00:00:00.000Z'
device[].dataRating[].dateTo	datetime string	The ending datetime for a data rating. <ul style="list-style-type: none"> If None, the data rating was still in affect when the request was made. Date Time format: <code>yyyy-MM-dd'T'HH:mm:ss.SSS'Z'</code>. datetime is represented in Coordinated Universal Time (UTC). 	dateTo: '2010-05-27T19:27:04.000Z'
device[].dataRating[].samplePeriod	double	The sample period, in seconds for the data rating.	samplePeriod: 1
device[].dataRating[].sampleSize	int	The size of the data sample for the data rating.	sampleSize: 1
device[].deviceCode	string	A code that unquely identifies a device.	deviceCode: 'BC_POD1_AD2M'
device[].deviceId	int	A numeric id that uniquely identifies a device.	deviceId: 11302
device[].deviceLink	string	The URL link to the Devices. Listing page for the specific device.	deviceLink: 'https://data.oceannetworks.ca/DeviceListing?DeviceId=11302'
device[].deviceName	string	The name of the device.	deviceName: 'Nortek Aquadopp HR-Profiler 2965'

Example - Print all devices deployed at Barkley Canyon - Axis, between June 1, 2016 and May 31, 2017

```
o = onc('YOUR_TOKEN_HERE');

devices = o.getDevices({'locationCode','BACAX', ...                               %BACAX - Barkley Canyon /
Axis
                                'dateFrom','2016-06-01T00:00:00.000Z', ...
                                'dateTo','2017-05-31T23:59:59.999Z'});

for i=1:numel(devices)
    disp(devices(i));
end
```

```
getDeployments(filters={})
```

Returns a list of deployments, as a list of structure objects that meet the criteria defined by the input filters.

Parameter	Type	Description	Example
Optional			
filters	struct or array	<p>A comma separated list of one or more filter criteria, used to return an exclusive set of deployment elements from the ONC deployments endpoint.</p> <ul style="list-style-type: none">• A list of filter/value pairs in the following format struct('<filter_1>','<value_1>','<filter_2>','<value_2>',...) or {'<filter_1>','<value_1>','<filter_2>','<value_2>',...}.• If excluded, all deployments are returned.• Valid filters<ul style="list-style-type: none">• locationCode, deviceCode, deviceCategoryCode, propertyCode, dateFrom, dateTo.• See devices Discovery Service for more information on filter usage.	{ 'deviceCode': 'NORTEK ADCP9917' }

Returns

A list of deployment objects.

```

    begin: '2016-06-14T08:58:41.000Z'
    depth: 982
    deviceCode: 'WFLNTU1087'
    end: []
  hasDeviceData: 1
    heading: []
    lat: 48.3166
    locationCode: 'BACAX'
    lon: -126.0508
    pitch: []
    roll: []

    begin: '2016-06-14T06:05:24.000Z'
    depth: 985
    deviceCode: 'BC_POD1_ROTSONAR'
    end: []
  hasDeviceData: 1
    heading: 90
    lat: 48.3166
    locationCode: 'BACAX'
    lon: -126.0508
    pitch: []
    roll: []

    begin: '2016-06-14T06:05:24.000Z'
    depth: 985
    deviceCode: 'NORTEKSIGNATURE550056'
    end: []
  hasDeviceData: 1
    heading: 340
    lat: 48.3166
    locationCode: 'BACAX'
    lon: -126.0508
    pitch: []
    roll: []

```

Property	Type	Description	Example
deployment[]			[]
deployment[].locationCode	string	The locationCode for the deployment location.	locationCode: "BACAX"
deployment[].deviceCode	string	The deviceCode for a deployed device.	deviceCode: "SBECTD19p6813"
deployment[].begin	datetime	The beginning datetime of the deployment. Date Time format: <code>yyyy-MM-dd'T'HH:mm:ss.SSS'Z'</code> . <ul style="list-style-type: none"> DateTime is represented in Coordinated Universal Time (UTC). 	dateFrom: '2010-07-27T00:00:00.000Z'
deployment[].end	datetime	The ending datetime of the deployment. <ul style="list-style-type: none"> If the deployment is current, the value is null. Date Time format: <code>yyyy-MM-dd'T'HH:mm:ss.SSS'Z'</code> . <ul style="list-style-type: none"> DateTime is represented in Coordinated Universal Time (UTC). 	dateTo: '2016-08-01T00:00:00.000Z'
deployment[].hasDeviceData	string	Indicates that data products can be requested using a device category code for the deployment.	hasDeviceData: 1
deployment[].depth	double	The depth of the device deployment.	depth: 982
deployment[].lat	double	The latitude of the device deployment.	lat: 48.31658
deployment[].lon	double	The longitude of the device deployment.	lon: -126.0508
deployment[].heading	double	The heading of the device deployment.	heading: 244

deployment[].pitch	double	The pitch of the device deployment.	pitch: null
deployment[].roll	double	The pitch of the device deployment.	roll: null

Example - Print all devices deployed at Barkley Canyon - Axis, between June 1, 2016 and May 31, 2017

```
o = onc('YOUR_TOKEN_HERE');

deployments = o.getDeployments({'locationCode':'BACAX','dateFrom':'2016-06-01T00:00:00.000Z','dateTo':'2017-05-31T23:59:59.999Z'}) #BACAX - Barkley Canyon / Axis

for i=1:numel(deployments)
    disp(deployments(i));
end
```

```
getDeviceCategories(filters)
```

Returns a list of deviceCategories, as a list of struct objects that meet the criteria defined by the input filters.

Parameter	Type	Description	Example
Optional			
filters	struct or array	<p>A comma separated list of one or more filter criteria, used to return an exclusive set of deviceCategory elements from the ONC deviceCategories endpoint.</p> <ul style="list-style-type: none"> A list of filter/value pairs in the following format struct (<filter_1>,<value_1>,<filter_2>,<value_2>,...) or {<filter_1>,<value_1>,<filter_2>,<value_2>,...}. If excluded, all device categories are returned. Valid filters <ul style="list-style-type: none"> deviceCategoryCode, deviceCategoryName, description, locationCode, propertyCode. See deviceCategories Discovery Service for more information on filter usage. 	<pre>struct ('deviceCategoryCode','HYD ROPHONE') or {'deviceCategoryCode','HYD ROPHONE'}</pre>

Returns

A list of device category objects.

```
cvTerm: [1x1 struct]
    deviceCategory: [1x1 struct]
uri: 'http://vocab.nerc.ac.uk/collection/L05/current/130/'
vocabulary: 'SeaDataNet device categories'

description: 'Conductivity Temperature (and Depth Sensor)'
deviceCategoryCode: 'CTD'
deviceCategoryName: 'CTD'
hasDeviceData: 'true'
longDescription: ' Conductivity Temperature Depth (CTD) is an abbreviated name for an instrument package that contains sensors for measuring the conductivity, temperature, and pressure of seawater. Salinity, sound velocity, and density are variables derived from sensor measurements. CTDs can carry additional instruments and sensors such as oxygen sensors, turbidity sensors and fluorometers. '
```

Property	Type	Description	Example
deviceCategory[]	list	A list of device categories.	

deviceCategory[].cvTerm	struct	A structure of controlled vocabulary terms.	cvTerm: [1x1 struct]
deviceCategory[].cvTerm. deviceCategory[]	struct	A list of structure of controlled vocabulary term information for the device category.	deviceCategory: [1x1 struct]
deviceCategory[].cvTerm. deviceCategory[].uri	string	The URI of the controlled vocabulary term.	uri: 'http://vocab.nerc.ac.uk/collection/L05/current/130/'
deviceCategory[].cvTerm. deviceCategory[].vocabulary	string	The name of the controlled vocabulary to which the term belongs.	vocabulary: 'SeaDataNet device categories'
deviceCategory[].description	string	The short description of the device category.	description: 'Conductivity Temperature (and Depth Sensor)'
deviceCategory[].deviceCategory Code	string	A code to uniquely identify a device category.	deviceCategoryCode: 'CTD'
deviceCategory[].deviceCategory Name	string	The name of the device category.	deviceCategoryName: 'CTD'
deviceCategory[].hasDeviceData	string	Indicates that data products can be requested using the device category for a location.	hasDeviceData: 'true'
deviceCategory[].longDescription	string	The long description of the device category.	longDescription: ' Conductivity Temperature Depth (CTD) is an abbreviated name for an instrument package that contains sensors for measuring the conductivity, temperature, and pressure of seawater. Salinity, sound velocity, and density are variables derived from sensor measurements. CTDs can carry additional instruments and sensors such as oxygen sensors, turbidity sensors and fluorometers. '

Example - Print all device categories available at Barkley Canyon - Upper Slope

```
o = ONC('YOUR_TOKEN_HERE');

deviceCategories = o.getDeviceCategories({'locationCode','NCBC'});    #NCBC - Barkely Canyon / Upper Slope

for i=1:numel(deviceCategories)
    disp(deviceCategories(i));
end
```

```
getProperties(filters)
```

Returns a list of properties, as a list of struct objects that meet the criteria defined by the input filters.

Parameter	Type	Description	Example
Optional			
filters	struct or array	<p>A comma separated list of one or more filter criteria, used to return an exclusive set of properties elements from the ONC properties endpoint.</p> <ul style="list-style-type: none"> A list of filter/value pairs in the following format struct ('<filter_1>','<value_1>','<filter_2>','<value_2>',...) or {'<filter_1>','<value_1>','<filter_2>','<value_2>',...}. If excluded, all properties are returned. Valid filters <ul style="list-style-type: none"> propertyCode, propertyName, description, locationCode, deviceCategoryCode, deviceCode. See properties Discovery Service for more information on filter usage. 	<pre>struct ('propertyCode','seawaterte mperature') or {'propertyCode','seawaterte mperature'}</pre>

Returns

A list of property objects.

```

        cvTerm: [1x1 struct]
                property: []
                    uom: [1x1 struct]
                                uri: 'http://vocab.nerc.ac.uk/collection/P06/current
/UPDB/'
                                vocabulary: 'BODC data storage units'

        description: 'Pressure'
        hasDeviceData: 'true'
        hasPropertyData: 'true'
        propertyCode: 'pressure'
        propertyName: 'Pressure'
        uom: 'decibar'
    
```

Property	Type	Description	Example
property[]	list	A list of properties.	
property[].cvTerm	struct	A structure of controlled vocabulary terms.	cvTerm: [1x1 struct]
property[].cvTerm. property[]	struct	A list of structures of controlled vocabulary term information for the property.	property: []
property[].cvTerm. property[].uri	string	The URI of the controlled vocabulary term.	
property[].cvTerm. property[].vocabulary	string	The name of the controlled vocabulary to which the term belongs.	
property[].cvTerm.uom[]	struct	A list of structures of controlled vocabulary term information for the unit of measure.	uom: [1x1 struct]
property[].cvTerm.uom[]. uri	string	The URI of the controlled vocabulary term.	uri: 'http://vocab.nerc.ac.uk /collection/P06/current/UPDB/'
property[].cvTerm.uom[]. vocabulary	string	The name of the controlled vocabulary to which the term belongs.	vocabulary: 'BODC data storage units'
property[].description	string	The short description of the device category.	description: 'Pressure'
property[].hasDeviceData	string	Indicates whether data products can be requested using the property code, along with a device category for a location.	hasDeviceData: 'true'
property[].hasPropertyData	string	Indicates whether data products can be requested using the property code for a location.	hasPropertyData: 'true'
property[].propertyCode	string	A code that uniquely identifies the property.	propertyCode: 'pressure'
property[].propertyName	string	The name of the property.	propertyName: 'Pressure'
property[].uom	string	The Unit of Measure used for the property measurements.	uom: 'decibar'

Example - Print all properties that are available at Barkley Canyon - Axis

```

o = onc('YOUR_TOKEN_HERE');

properties = o.getProperties({'locationCode','BACAX'}); #BACAX - Barkley Canyon / Axis

for i=1:numel(properties)
    disp(properties(i));
end
    
```

```
getDataProducts(filters)
```

Returns a list of data products as a list of struct objects that meet the criteria defined by the input filters.

Parameter	Type	Description	Example
Optional			
filters	struct or array	<p>A comma separated list of one or more filter criteria, used to return an exclusive set of data product & extension elements from the ONC dataProducts endpoint.</p> <ul style="list-style-type: none">• A list of filter/value pairs in the following format struct('<filter_1>','<value_1>','<filter_2>','<value_2>',...)• or {'<filter_1>','<value_1>','<filter_2>','<value_2>',...}.• If excluded, all data products are returned.• Valid filters<ul style="list-style-type: none">• dataProductCode, extension, locationCode, deviceCategoryCode, deviceCode, propertyCode.• See dataProducts Discovery Service for more information on filter usage.	<pre>struct ('dataProductCode', 'TSS D', ... 'extension', 'json') or {'dataProductCode', 'TSS D', ... 'extension', 'json'}</pre>

Returns

A list of data product objects.

```
dataProductCode: 'TSSD'
dataProductName: 'Time Series Scalar Data'
  extension: 'json'
  hasDeviceData: 1
  hasPropertyData: 1
  helpDocument: 'https://wiki.oceannetworks.ca/display/DP/1'
```

Property	Type	Description	Example
dataProduct[]	list	A list of data products.	[]
dataProduct[]. dataProductCode	string	A code that uniquely identifies a data product.	dataProductCode: 'TSSD'
dataProduct[]. dataProductName	string	The name of the data product.	dataProductName: 'Time Series Scalar Data'
dataProduct[]. extension	string	The file name extension for a data product.	extension: 'json'
dataProduct[]. hasDeviceData	string	Indicates whether the data product can be requested using a device category code and a location code.	hasDeviceData: 1
dataProduct[]. hasPropertyData	string	Indicates whether the data product can be requested using a property code and a location code or device code.	hasPropertyData: 1
dataProduct[]. helpDocument	string	A Link URL to the Oceans 2.0 Help documentation for the specific Data Product.	helpDocument: 'https://wiki.oceannetworks.ca/display/DP/1'

Example - Print all MatLab data product

```
o = onc('YOUR_TOKEN_HERE');

dataProducts = o.getDataProducts({'extension','mat'}); %mat - MatLab

for i=1:numel(dataProducts)
    disp(dataProducts(i));
end
```

Data Product Download methods

```
orderDataProduct(filters, maxRetries, downloadResultsOnly, includeMetadataFile)
```

Requests a data product and downloads the generated files to the class's outPath.

This method performs the complete data product download workflow by calling additional methods to perform the following steps:

1. Request a data product, which includes estimates for download size and time.
2. Run a requested data product, which kicks off a process on the task machine to generate the data product.
3. Download the data product to disk. The process will continue to poll the web service to determine if the product is ready to download. If it is not ready, status messages will be provided. Once it is ready, it will download the data product to disk.

Parameter	Type	Description	Example
Required			
filters	struct or array	<p>A comma separated list of filter criteria parameters, used to request data as a data product from the ONC dataProductDelivery endpoint.</p> <ul style="list-style-type: none">• A list of filter/value pairs in the following format struct('<filter_1>','<value_1>','<filter_2>','<value_2>') or {'<filter_1>','<value_1>','<filter_2>','<value_2>',...}.• Required filters<ul style="list-style-type: none">• For Site Device data request include locationCode and deviceCategoryCode.• For Primary Sensor level data request include locationCode and propertyCode.• For Sensor Level data request include locationCode, deviceCategoryCode and propertyCode.• For Device (all properties) data request use deviceCode.• For Device Property data request use deviceCode and deviceCategoryCode.• Include dateFrom and dateTo (a date range).• Include dataProductCode, extension and appropriate data product options.• See locations Discovery Service, devices Discovery Service, deviceCategories Discovery Service, properties Discovery Service and dataProducts Discovery Service for available parameters (filters).	<pre>struct ('locationCode','BACAX',... 'deviceCategoryCode','ADCP2MHZ',... 'dataProductCode','TS SD',... 'extension','csv',... 'dateFrom','2016-07-27T00:00:00.000Z',... 'dateTo','2016-08-01T00:00:00.000Z',... 'dpo_qualityControl',1,... 'dpo_resample','none',... 'dpo_dataGaps',0) or {'locationCode','BACAX',... 'deviceCategoryCode','ADCP2MHZ',... 'dataProductCode','TS SD',... 'extension','csv',... 'dateFrom','2016-07-27T00:00:00.000Z',... 'dateTo','2016-08-01T00:00:00.000Z',... 'dpo_qualityControl',1,... 'dpo_resample','none',... 'dpo_dataGaps',0}</pre>

Option			
maxRetries	int	The number of times to retry the service before the function aborts.	
		<ul style="list-style-type: none"> If excluded, the default of 100 is used. 	
downloadResultsOnly	boolean	Indicates whether the files will be downloaded or if only the URL to the file will be returned.	
		<ul style="list-style-type: none"> false - Files are downloaded to the ONC class outPath property location. true - Files will not be downloaded. The download URLs are available from the url property for each downloadResult in downloadResults in the returned struct. If excluded, the default of false is used. 	
includeMetadataFile	boolean	Indicates whether the metadata file associated with the data product request will be downloaded.	
		<ul style="list-style-type: none"> false - Metadata file is not downloaded. true - Metadata file is downloaded. If excluded, the default of false is used. 	

Returns

A structure of download results and run time statistics.

```

url: "https://data.oceannetworks.ca/api/dataProductDelivery?method=download&token=b6ede000-1865-4ac3-94ad-e87d8bdfd307&dpRunId=4658962&index=1"
messages: {1x2 cell}
status: 'complete'
size: 3060227
file: 'c:\ONC\data\BarkleyCanyon_Axis_ADCP2MHz_20160727T000005Z_20160731T235958Z-clean.csv'
index: 1
downloaded: 1

url: "https://data.oceannetworks.ca/api/dataProductDelivery?method=download&token=b6ede000-1865-4ac3-94ad-e87d8bdfd307&dpRunId=4658962&index=meta"
messages: {}
status: 'complete'
size: 27504
file: 'c:\ONC\data\BarkleyCanyon_Axis_ADCP2MHz_20160727T000005Z_20160731T235958Z-clean_CSV_META.pdf'
index: 'meta'
downloaded: 1

```

Property	Type	Description
downloadResults	list	A cell array of download results struct objects.
downloadResults{}.url	string	The URL used to make the download request.
downloadResults{}.messages	list	A cell array of the messages that were returned from the dataProductDelivery download method when it was polled.
downloadResults{}.status	string	The status of the download process at the request. Possible values are: <ul style="list-style-type: none"> "running" - The data product request is running on the Task server. "complete" - The data product request Task has completed and file has been downloaded or URL has been returned. "error" - An error, preventing the task execution or download from completing successfully, has occurred.
downloadResults{}.size	float	The actual size of the file in bytes.
downloadResults{}.file	string	The full path of the file that was downloaded.
downloadResults{}.index	string	The index of the file downloaded. The index can either be a number 1-* or "meta" for the data product metadata file.
downloadResults{}.downloaded	boolean	Indicates whether the file was downloaded. If the filter downloadResultsOnly=True is used, the data product is not downloaded and the value will be False.

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

```
o = onc('YOUR_TOKEN_HERE');

results = o.orderDataProduct(struct('locationCode','BACAX',...
                                   'deviceCategoryCode','ADCP2MHZ',...
                                   'dataProductCode','TSSD',...
                                   'extension','csv',...
                                   'dateFrom','2016-07-27T00:00:00.000Z',...
                                   'dateTo','2016-08-01T00:00:00.000Z',...
                                   'dpo_qualityControl',1,...
                                   'dpo_resample','none',...
                                   'dpo_dataGaps',0),100,false,true);

for i=1:numel(results)
    disp(results{i});
end
```

```
requestDataProduct(filters)
```

Requests a data product using the dataProductDelivery service and provides request information so that the calling function can decide whether to run the data product.

The struct of information it returns includes a requestID, which is used to run the data product, and estimates of the expected download times and file sizes.

Parameter	Type	Description	Example
Required			

filters	struct or array	<p>A comma separated list of filter criteria parameters, used to request data as a data product from the ONC dataProductDelivery endpoint.</p> <ul style="list-style-type: none"> A list of filter/value pairs in the following format struct('<filter_1>','<value_1>','<filter_2>','<value_2>',...)' or {'<filter_1>','<value_1>','<filter_2>','<value_2>',...}. Required filters <ul style="list-style-type: none"> For Site Device data request include locationCode and deviceCategoryCode. For Primary Sensor level data request include locationCode and propertyCode. For Sensor Level data request include locationCode, deviceCategoryCode and propertyCode. For Device (all properties) data request use deviceCode. For Device Property data request use deviceCode and deviceCategoryCode. Include dateFrom and dateTo (a date range). Include dataProductCode, extension and appropriate data product options. See locations Discovery Service, devices Discovery Service, deviceCategories Discovery Service, properties Discovery Service and dataProducts Discovery Service for available parameters (filters). 	<pre> struct ('locationCode', 'BACAX', ... 'deviceCategoryCode', 'AD CP2MHZ', ... 'dataProductCode', 'TSSD' ,... 'extension', 'csv', ... 'dateFrom', '2016- 07-27T00:00:00.000Z', ... 'dateTo', '2016- 08-01T00:00:00.000Z', ... 'dpo_qualityControl', 1, ... 'dpo_resample', 'none', ... 'dpo_dataGaps', 0) or {'locationCode', 'BACAX', ... 'deviceCategoryCode', 'AD CP2MHZ', ... 'dataProductCode', 'TSSD' ,... 'extension', 'csv', ... 'dateFrom', '2016-07- 27T00:00:00.000Z', ... 'dateTo', '2016-08- 01T00:00:00.000Z', ... 'dpo_qualityControl', 1, ... 'dpo_resample', 'none', ... 'dpo_dataGaps', 0} </pre>
---------	-----------------	---	--

Returns

A data product request structure

for archived files (AD):

```

dpRequestId: 2046404
compressedFileSize: 12563408
  fileSize: 70766230
  numFiles: 4
downloadTimes: {1x3 cell}

10Mbps: 7.076623
50Mbps: 1.4153247
150Mbps: 0.47177488

```

for generated files:

```

dpRequestId: 2046404
estimatedFileSize: 8348187
estimatedProcessingTime: 16

```

Property	Type	Description	Example
Always contains			
dpRequestId	int	A unique id for a data product request.	dpRequestId: 2046404

May contain			
compressedFileSize	int	The compressed size of the known file(s) in bytes.	compressedFileSize: 12563408
fileSize	int	The size of known file(s) in bytes.	fileSize: 70766230
numFiles	int	The number of files.	numFiles: 4
downloadTimes	cell array	A structure of estimated download times.	downloadTimes: {1x3 cell} 10Mbps: 7.076623 50Mbps: 1.4153247 150Mbps: 0.47177488
estimatedFileSize	int	The estimated file size of the generated data product.	estimatedFileSize: 8348187
estimatedProcessingTime	int	The estimated time, in seconds, that it will take to run the data product request.	estimatedProcessingTime: 16

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

```
o = onc('YOUR_TOKEN_HERE');

requestInfo = o.requestDataProduct(struct('locationCode','BACAX',...
                                         'deviceCategoryCode','ADCP2MHZ',...
                                         'dataProductCode','TSSD',...
                                         'extension','csv',...
                                         'dateFrom','2016-07-27T00:00:00.000Z',...
                                         'dateTo','2016-08-01T00:00:00.000Z',...
                                         'dpo_qualityControl',1,...
                                         'dpo_resample','none',...
                                         'dpo_dataGaps',0));

disp(requestInfo);
```

```
runDataProduct(requestId)
```

Runs a data product request using the dataProductDelivery service with a Request Id and returns a list of runIds

Parameter	Type	Description	Example
Required			
requestId	int	A dataProductDelivery dpRequestId, which is returned by the dataProductDelivery service request method, or as the 'dpRequestId' property in the dpRequest returned by the requestDataProduct() method.	2046404

Returns

A list of runIds.

```
dpRunId: 4659102
status: 'data product running'
```

Example - Run a requested data product

```
o = onc('YOUR_TOKEN_HERE');  
  
runs = o.runDataProduct(YOUR_REQUEST_ID_HERE);  
  
disp(runs);
```

```
downloadDataProduct(runId, fileCount, estimatedProcessingTime, maxRetries, downloadResultsOnly,  
includeMetadataFile)
```

Polls the dataProductDelivery service until data product generation task is complete and downloads all of the files and/or retrieves the file information (URL, file and download status).

Parameter	Type	Description	Example
Required			
runId	int	A dataProductDelivery dpRunId, which is returned by the dataProductDelivery service run method, or as the dpRunId property in the list returned by the runDataProduct() method.	4629218
Option			
fileCount	int	The actual or estimated file count, which is returned from the dataProductDelivery request method. <ul style="list-style-type: none">If there is no estimated file count, use 0.If excluded, the default of 0 is used.	1
estimatedProcessingTime	int	The estimated time in seconds, to run the request on the task machine. <ul style="list-style-type: none">Used to determine how often the dataProductDelivery service is called to determine whether the task processing is complete.Estimated processing time is provided by the dataProductDelivery request method.	2
maxRetries	int	The number of times to retry the service before the function aborts. <ul style="list-style-type: none">If excluded, the default of 100 is used.	10
downloadResultsOnly	boolean	Determines whether the files will be downloaded or if only the URL to the file will be returned. <ul style="list-style-type: none">false - Files are downloaded to the ONC class outPath property location.true - Files will not be downloaded. The download URLs are available from the url property for each downloadResult in downloadResults in the returned struct.If excluded, the default of false is used.	true
includeMetadataFile	boolean	Indicates whether the metadata file associated with the data product request will be downloaded. <ul style="list-style-type: none">false - Metadata file is not downloaded.true - Metadata file is downloaded.If excluded, the default of false is used.	true

Returns

A list of download results struct objects.

```

url: "https://data.oceannetworks.ca/api/dataProductDelivery?
method=download&token=<YOUR_TOKEN>&dpRunId=<YOUR_RUN_ID>&index=1"
messages: {1x2 cell}
status: 'complete'
size: 3060227
file: 'c:\ONC\data\BarkleyCanyon_Axis_ADCP2MHz_20160727T000005Z_20160731T235958Z-clean.csv'
index: 1
downloaded: 1

url: "https://data.oceannetworks.ca/api/dataProductDelivery?
method=download&token=<YOUR_TOKEN>&dpRunId=<YOUR_RUN_ID>&index=meta"
messages: {}
status: 'complete'
size: 27504
file: 'c:\ONC\data\BarkleyCanyon_Axis_ADCP2MHz_20160727T000005Z_20160731T235958Z-clean_CSV_META.pdf'
index: 'meta'
downloaded: 1

```

Property	Type	Description
url	string	The URL used to make the download request.
message	list	A list of the messages that were returned from the dataProductDelivery download method when it was polled.
status	string	The status of the download process at the request. Possible values are: <ul style="list-style-type: none"> "running" - The data product request is running on the Task server. "complete" - The data product request Task has completed and file has been downloaded or URL has been returned. "error" - An error, preventing the task execution or download from completing successfully, has occurred.
size	float	The actual size of the file in bytes.
file	string	The full path of the file that was downloaded.
index	string	The index of the file downloaded. The index can either be a number 1-* or "meta" for the data product metadata file.
downloaded	boolean	Indicates whether the file was downloaded. If the filter downloadResultsOnly=True is used, the data product is not downloaded and the value will be False

Example - Download the results of requested data product run

```

o = onc('YOUR_TOKEN_HERE');

results = o.downloadDataProduct(YOUR_RUN_ID_HERE, 1);

for i=1:numel(results)
    disp(results{i});
end

```

```

getDataProductUrls(filters, maxRetries=100)

```

Orders a data product and return only the URLs, for download at a later time. URLs can be used with the downloadFile() or getJsonFromUrl() methods.

Parameter	Type	Description	Example
Required			

filters	struct or array	<p>A comma separated list of filter criteria parameters, used to request data as a data product from the ONC dataProductDelivery endpoint.</p> <ul style="list-style-type: none"> A list of filter/value pairs in the following format struct ('<filter_1>','<value_1>','<filter_2>','<value_2>',...) or {'<filter_1>','<value_1>','<filter_2>','<value_2>',...}. Required filters <ul style="list-style-type: none"> For Site Device data request include locationCode and deviceCategoryCode. For Primary Sensor level data request include locationCode and propertyCode. For Sensor Level data request include locationCode, deviceCategoryCode and propertyCode. For Device (all properties) data request use deviceCode. For Device Property data request use deviceCode and deviceCategoryCode. Include dateFrom and dateTo (a date range). Include dataProductCode, extension and appropriate data product options. See locations Discovery Service, devices Discovery Service, deviceCategories Discovery Service, properties Discovery Service and dataProducts Discovery Service for available parameters (filters). 	<pre> struct ('locationCode', 'BACAX', ... 'deviceCategoryCode', 'AD CP2MHZ', ... 'dataProductCode', 'TSSD' ,... 'extension', 'csv', ... 'dateFrom', '2016- 07-27T00:00:00.000Z', ... 'dateTo', '2016- 08-01T00:00:00.000Z', ... 'dpo_qualityControl', 1, ... 'dpo_resample', 'none', ... 'dpo_dataGaps', 0) or {'locationCode', 'BACAX', ... 'deviceCategoryCode', 'AD CP2MHZ', ... 'dataProductCode', 'TSSD' , ... 'extension', 'csv', ... 'dateFrom', '2016-07- 27T00:00:00.000Z', ... 'dateTo', '2016-08- 01T00:00:00.000Z', ... 'dpo_qualityControl', 1, ... 'dpo_resample', 'none', ... 'dpo_dataGaps', 0} </pre>
Option			
maxRetries	int	<p>The number of times to retry the service before the function aborts.</p> <ul style="list-style-type: none"> If excluded, the default of 100 is used. 	10

Returns

A list of urls.

```
[ "https://data.oceannetworks.ca/api/dataProductDelivery?
method=download&token=<YOUR_TOKEN>&dpRunId=<YOUR_RUN_ID>&index=1" ]
```

Example - Order a data product and get a list of the download urls

```
o = onc('YOUR_TOKEN_HERE');

urls = o.getDataProductUrls(struct('locationCode','BACAX',...
                                  'deviceCategoryCode','ADCP2MHZ',...
                                  'dataProductCode','TSSD',...
                                  'extension','json',...
                                  'dateFrom','2016-07-27T00:00:00.000Z',...
                                  'dateTo','2016-08-01T00:00:00.000Z',...
                                  'dpo_qualityControl',1,...
                                  'dpo_resample','none',...
                                  'dpo_dataGaps',0,...
                                  'dpo_jsonOutputEncoding','OM'),100);

for i=1:numel(urls)
    disp(urls{i});
end
```

```
downloadFile(url, file=[])
```

Downloads a file from a URL, write it to a file and return download results information (URL, file, message and download status).

Parameter	Type	Description	Example
Required			
url	string	The URL to be downloaded.	<code>https://data.oceannetworks.ca/api/dataProductDelivery?method=download&token=<YOUR_TOKEN_HERE>&dpRunId=<YOUR_RUN_ID_HERE>&index=<YOUR_INDEX_HERE></code>
file	string	The full path of the file download destination. <ul style="list-style-type: none">If excluded, the file name from the content disposition in the header is used, along with the onc class object outPath to create a full file name path.	<code>c:/temp/myDownload.csv</code>

Returns

A downloadResults structure.

```
url: "https://data.oceannetworks.ca/api/dataProductDelivery?method=download&token=b6ede000-1865-4ac3-94ad-e87d8bdfd307&dpRunId=4722272&index=1"
messages: {}
file: 'c:\ONC\data\matlab\myDownload.json'
downloaded: 1
status: 'complete'
```

Property	Type	Description
url	string	The URL used to make the download request.
file	string	The full path of the file that was downloaded.

message	list	A list of the messages that were returned from the dataProductDelivery download method when it was polled.
downloaded	boolean	Indicates whether the file was downloaded.
status	string	The status of the download process at the request. Possible values are: <ul style="list-style-type: none"> • "running" - The data product request is running on the Task server. • "complete" - The data product request Task has completed and file has been downloaded or URL has been returned. • "error" - An error, preventing the task execution or download from completing successfully, has occurred.

Example - Download a file from a url

```
o = onc('YOUR_TOKEN_HERE');

url = 'https://data.oceannetworks.ca/api/dataProductDelivery?
method=download&token=<YOUR_TOKEN_HERE>&dpRunId=<YOUR_RUN_ID_HERE>&index=<YOUR_INDEX_HERE>';
file = 'c:/temp/myDownload.csv';

downloadResult = o.downloadFile(url, file);

disp(downloadResult);
```

```
decodeJsonFromUrl(url)
```

Returns a structure object from the JSON returned from a URL

Parameter	Type	Description	Example
Required			
url	string	The URL to a JSON data product delivery result.	https://data.oceannetworks.ca/api/dataProductDelivery?method=download&token=<YOUR_TOKEN_HERE>&dpRunId=<YOUR_RUN_ID_HERE>&index=<YOUR_INDEX_HERE>

Returns

A structure encoded from a JSON string.

Example - Get dictionary of results from json data product request

```

o = onc('YOUR_TOKEN_HERE');

urls = o.getDataProductUrls(struct('locationCode','BACAX',...
                                'deviceCategoryCode','ADCP2MHZ',...
                                'dataProductCode','TSSD',...
                                'extension','json',...
                                'dateFrom','2016-07-27T00:00:00.000Z',...
                                'dateTo','2016-08-01T00:00:00.000Z',...
                                'dpo_qualityControl',1,...
                                'dpo_resample','none',...
                                'dpo_dataGaps',0,...
                                'dpo_jsonOutputEncoding','OM'),100);

for i=1:numel(urls)
    object = o.decodeJsonFromUrl(url);
    disp(object);
end

```

Near Real-Time data product methods

getDirectScalarByStation(filters, outputFormat, metadata, rowLimit) - deprecated in favour of getDirectScalar()

Returns scalar data, in JSON Object or Array format, in the response payload that meet the criteria defined by the input filters.

Parameter	Type	Description	Example
Required			
filters	struct or array	<p>A comma separated list of filter criteria parameters, used to request data and metadata from the scalar data endpoint.</p> <ul style="list-style-type: none"> A list of filter/value pairs in the following format struct('<filter_1>','<value_1>','<filter_2>','<value_2>',...). or {'<filter_1>','<value_1>','<filter_2>','<value_2>',...}. Filters must include locationCode and deviceCategoryCode. Filters can optionally include begin and end (a date range). See locations Discovery Service and deviceCategories Discovery Service for more information on filter usage. 	<pre> struct ('locationCode','TWDP', ... 'deviceCategoryCode','TS G', ... 'begin','2016-09-01T00:00: 00.000Z', ... 'end','2016-09-01T08:00: 00.000Z') or {'locationCode','TWDP', ... 'deviceCategoryCode','TS G', ... 'begin','2016-09-01T00:00: 00.000Z', ... 'end','2016-09-01T08:00: 00.000Z'} </pre>
Option			
outputFormat	string	<p>The encoding of the response output.</p> <ul style="list-style-type: none"> Object - Data are a list of structures with sampleTime, value and qaqcFlag for a given sensor. Array - Data have 3 arrays for a given sensor. Array of sampleTime, array of values and array of qaqcFlags. If excluded, the default 'Object' is used. 	"outputFormat","Array"
metadata	string	<p>The amount of metadata detail included in the response output.</p> <ul style="list-style-type: none"> Minimum - Provides only basic property information. Response includes serviceMetadata structure only. Full - Provides all property information. Response includes both metadata and serviceMetadata structures. If excluded, the default 'Minimum' is used. 	"metadata","Full"

rowLimit	integer	The maximum number of rows of scalar data that will be returned for each sensor. <ul style="list-style-type: none"> Maximum is 100,000. If excluded, the default 100,000 is used. 	"rowLimit",5000
----------	---------	---	-----------------

Returns

A structure that contains sensor data that matches the filter criteria defined in the input parameters, metadata and service metadata.

```

metadata: [1x1 struct]
    bbox: [1x1 struct]
        maxDepth: 3
        maxLat: 49.1139
        maxLon: -123.5264
        minDepth: 3
        minLat: 49.1139
        minLon: -123.5264
    depth: 3
    deviceCategoryName: 'TSG'
    latitude: 49.1139
    longitude: -123.5264
    begin: '2016-09-01T00:00:00.000Z'
    end: '2016-09-01T00:00:30.000Z'
    locationCode: 'TWDP'
sensorData: [3x1 struct]
    actualSamples: 1
    data: [1x1 struct]
        qaqcFlag: 1
        sampleTime: '2016-09-01T00:00:07.037Z'
        value: 3.4429
    sensor: 'Conductivity'
    sensorName: 'Conductivity'
    unitOfMeasure: 'S/m'
serviceMetadata: [1x1 struct]
    metadata: 'Full'
    nextDateFrom: '2016-09-01T00:00:17.041Z'
    outputFormat: 'Object'
    rowLimit: 100000
    sensors: []
    totalActualSamples: 3
    begin: '2016-09-01T00:00:00.000Z'
    end: '2016-09-01T00:00:10.000Z'
    deviceCategoryCode: 'TSG'
    locationCode: 'TWDP'

```

Property	Type	Description
metadata	struct	A structure of metadata, if metadata='Full' input parameter is used.
metadata.bbox	struct	A Bounding Box structure.
metadata.bbox.maxDepth	float	The maximum depth of the instruments contributing to the results.
metadata.bbox.maxLat	float	The maximum Latitude of bounding box of the instruments contributing to the results.
metadata.bbox.maxLon	float	The maximum Longitude of the bounding box of the instruments contributing to the results.
metadata.bbox.minDepth	float	The minimum Depth of the instruments contributing to the results.
metadata.bbox.minLat	float	The minimum Latitude of bounding box of the instruments contributing to the results.
metadata.bbox.minLon	float	The minimum Longitude of the bounding box of the instruments contributing to the results.
sensorData	list	A list of sensor structures.

sensorData[].actualSamples	int	The number of samples in the data list.
sensorData[].data	list	A list of observation structures.
sensorData[].data[].qaqcFlag	int	The QAQC Flag for the observation.
sensorData[].data[].sampleTime	datetime string	The recorded time of the observation. <ul style="list-style-type: none"> • Date Time format: <code>yyyy-MM-ddT'HH:mm:ss.SSS'Z'</code>. • datetime is represented in Coordinated Universal Time (UTC).
sensorData[].data[].value	float	The value of the observation.
sensorData[].sensor	string	The code of the sensor that is being observed.
sensorData[].sensorName	string	The name of the sensor that is being observed.
sensorData[].unitOfMeasure	string	The unit of measure of the observation.
serviceMetadata	struct	A structure of the metadata for the service request.
serviceMetadata.metadata	string	The requested metadata filter.
serviceMetadata.nextDataFrom	datetime string	The begin date for next reading in <code>yyyy-MM-ddTHH:mm:ss.SSSZ</code> format.
serviceMetadata.outputFormat	string	The requested JSON output format.
serviceMetadata.rowLimit	int	The requested row limit per sensor.
serviceMetadata.sensors	string	A comma separated list of the requested sensors.
serviceMetadata.totalActualSamples	int	The number of observations expected for all specified sensors.
serviceMetadata.begin	datetime string	The begin date of the request in <code>yyyy-MM-ddTHH:mm:ss.SSSZ</code> format.
serviceMetadata.end	datetime string	The end date of the request in <code>yyyy-MM-ddTHH:mm:ss.SSSZ</code> format.
serviceMetadata.deviceCategoryCode	string	The Device Category Code of the request. See Available Device Categories for further details.
serviceMetadata.locationCode	string	The Location Code of the request. See Available Locations for further details.

Example - Print the last Thermosalinograph reading from Tswassen - Duke Point Ferry

```
o = onc('YOUR_TOKEN_HERE');

result = o.getDirectScalarByStation({'locationCode','TWDP','deviceCategoryCode','TSG'});

disp(result);
```

Example - Print 1 hour of Thermosalinograph readings from Tswassen - Duke Point Ferry

```
o = onc('YOUR_TOKEN_HERE');

result = o.getDirectScalarByStation(struct('locationCode','TWDP','deviceCategoryCode','TSG','begin','2016-09-01T00:00:00.000Z','end','2016-09-01T01:00:00.000Z'));

disp(result);
```

Example - Print 1 hour of Thermosalinograph readings from Tswassen - Duke Point Ferry as an Array

```
o = onc('YOUR_TOKEN_HERE');

result = o.getDirectScalarByStation(struct('locationCode':'TWDP','deviceCategoryCode':'TSG','begin':'2016-09-01T00:00:00.000Z','end':'2016-09-01T01:00:00.000Z'),'Array');

disp(result);
```

```
getDirectScalar(filters)
```

Returns scalar data in the response payload that meet the criteria defined by the input filters.

Parameter	Type	Description	Example
Required			
filters	struct or array	<p>A comma separated list of filter criteria parameters, used to request data and metadata from the scalardata endpoint.</p> <ul style="list-style-type: none">• A list of filter/value pairs in the following format struct (<filter_1>,<value_1>,<filter_2>,<value_2>,...) or {<filter_1>,<value_1>,<filter_2>,<value_2>,...}.• Filters must include locationCode and deviceCategoryCode.• Filters can optionally include dateFrom and dateTo (a date range).• Filters can optionally specify "metadata" as "minimum" (default) or "full".<ul style="list-style-type: none">• Minimum provides only basic property information.• Full provides all property information. Response includes metadata structure.• Filters can optionally specify "rowLimit", the maximum number of rows of data that will be returned for each sensor.<ul style="list-style-type: none">• The maximum is 100,000. If omitted or a higher or invalid limit is specified, the default of 100,000 is used.• See locations Discovery Service and deviceCategories Discovery Service for more information on filter usage.	<pre>struct('locationCode','TWDP', ... 'deviceCategoryCode','TSG', ... 'dateFrom','2016-09-01T00:00:00.000Z', ... 'dateTo','2016-09-01T08:00:00.000Z') or {'locationCode','TWDP', ... 'deviceCategoryCode','TSG', ... 'dateFrom','2016-09-01T00:00:00.000Z', ... 'dateTo','2016-09-01T08:00:00.000Z'}</pre>

Returns

A structure that contains sensor data that matches the filter criteria defined in the input parameters, metadata and service metadata.

```

metadata: [1x1 struct]
    boundingBox: [1x1 struct]
        maxDepth: 3
        maxLat: 49.1139
        maxLon: -123.5264
        minDepth: 3
        minLat: 49.1139
        minLon: -123.5264
    depth: 3
    deviceCategoryCode: 'Thermosalinograph'
    latitude: 49.1139
    longitude: -123.5264
    locationName: 'Tsawwassen - Duke Point'
sensorData: [3x1 struct]
    actualSamples: 1
    data: [1x1 struct]
        qaqcFlag: 1
        sampleTime: '2016-09-01T00:00:07.037Z'
        value: 3.4429
        sensor: 'Conductivity'
        sensorName: 'Conductivity'
        unitOfMeasure: 'S/m'

queryURL: 'https://data.oceannetworks.ca/api/scalardata?method=getByLocation&token=f68bc7e1-f231-481d-b32e-590c5c1e1ee8&locationCode=TWDP&d...' <Preview truncated at 128 characters>

next: [1x1 struct]
    url: 'https://data.oceannetworks.ca/api/scalardata?metadata=full&method=getByLocation&end=2016-08-01T00%3A00%3A00.000Z&rowLimit=10&loc...' <Preview truncated at 128 characters>
    parameters: [1x1 struct]
        metadata: 'Full'
        token: 'YOUR_TOKEN_HERE'
        rowLimit: 10
        begin: '2016-09-01T00:00:00.000Z'
        end: '2016-09-01T00:00:10.000Z'
    deviceCategoryCode: 'TSG'
    locationCode: 'TWDP'

```

Property	Type	Description
metadata	struct	A structure of metadata, if metadata='Full' input parameter is used.
metadata.boundingBox	struct	A Bounding Box structure.
metadata.boundingBox.maxDepth	float	The maximum depth of the instruments contributing to the results.
metadata.boundingBox.maxLat	float	The maximum Latitude of bounding box of the instruments contributing to the results.
metadata.boundingBox.maxLon	float	The maximum Longitude of the bounding box of the instruments contributing to the results.
metadata.boundingBox.minDepth	float	The minimum Depth of the instruments contributing to the results.
metadata.boundingBox.minLat	float	The minimum Latitude of bounding box of the instruments contributing to the results.
metadata.boundingBox.minLon	float	The minimum Longitude of the bounding box of the instruments contributing to the results.
sensorData	list	A list of sensor structures.
sensorData[].actualSamples	int	The number of samples in the data list.
sensorData[].data	struct	A structure of observation data lists.
sensorData[].data.qaqcFlags	list of int	The QAQC Flag for the observation.

sensorData[].data.sampleTimes	list of datetime string	The recorded time of the observation. <ul style="list-style-type: none"> Date Time format: <code>yyyy-MM-ddT'HH:mm:ss.SSS'Z'</code>. datetime is represented in Coordinated Universal Time (UTC).
sensorData[].data.values	list of float	The value of the observation.
sensorData[].sensorCode	list	The code of the sensor that is being observed.
sensorData[].sensorName	string	The name of the sensor that is being observed.
sensorData[].unitOfMeasure	string	The unit of measure of the observation.
next	struct	A structure of the metadata for the service request.
next.url	string	The request url.
next.parameters	struct	A structure of the metadata for the service request.
next.parameters.rowLimit	int	The requested row limit per sensor.
next.parameters.token	string	The user's token.
next.parameters.method	string	'getByLocation'.
next.parameters.begin	datetime string	The begin date of the request in <code>yyyy-MM-ddTHH:mm:ss.SSSZ</code> format.
next.parameters.end	datetime string	The end date of the request in <code>yyyy-MM-ddTHH:mm:ss.SSSZ</code> format.
next.parameters.deviceCategoryCode	string	The Device Category Code of the request. See Available Device Categories for further details.
next.parameters.locationCode	string	The Location Code of the request. See Available Locations for further details.

Example - Print the last Thermosalinograph reading from Tswassen - Duke Point Ferry

```
o = onc('YOUR_TOKEN_HERE');
result = o.getDirectScalar({'locationCode','TWDP','deviceCategoryCode','TSG'});
disp(result);
```

Example - Print 1 hour of Thermosalinograph readings from Tswassen - Duke Point Ferry

```
o = onc('YOUR_TOKEN_HERE');
result = o.getDirectScalar(struct('locationCode','TWDP','deviceCategoryCode','TSG','dateFrom','2016-09-01T00:00:00.000Z','dateTo','2016-09-01T01:00:00.000Z'));
disp(result);
```

`getDirectRawByStation(filters, rowLimit)` - deprecated in favour of `getDirectRawByLocation()`

Returns raw data from an instrument in the payload, in JSON format that meet the criteria defined by the input filters.

Parameter	Type	Description	Example
Required			

filter	struct or cell array	A comma separated list of filter criteria parameters, used to request data and metadata from the rawdata endpoint. <ul style="list-style-type: none"> A list of filter/value pairs in the following format struct('<filter_1>','<value_1>','<filter_2>','<value_2>',...) or {'<filter_1>','<value_1>','<filter_2>','<value_2>',...}. Filters must include locationCode and deviceCategoryCode. Filters can optionally include begin and end (a date range). See locations Discovery Service and deviceCategories Discovery Service for more information on filter usage. 	struct('locationCode','TWD P', ... 'deviceCategoryCode','TS G', ... 'begin','2016-09-01T00:00:00.000Z', ... 'end','2016-09-01T08:00:00.000Z') or {'locationCode','TWDP', ... 'deviceCategoryCode','TSG', ... 'begin','2016-09-01T00:00:00.000Z', ... 'end','2016-09-01T08:00:00.000Z'}
Option			
rowLimit	integer	The maximum number of rows of raw data that will be returned <ul style="list-style-type: none"> Maximum is 100,000. If excluded, the default 100,000 is used. 	"rowLimit",5000

Returns

A structure that contains sensor data matching the filter criteria defined in the input parameters, metadata and service metadata.

```

data: [1x1 struct]
    rawData: '20160901-00:00:07.037, 12.9162, 3.44294, 28.9096'
    sampleTime: '2016-09-01T00:00:07.037Z'

metadata: [1x1 struct]
    dataMetadata: [1x1 struct]
        firstTimestamp: '2016-09-01T00:00:07.037Z'
        hasMoreDataInRange: 0
        lastTimestamp: '2016-09-01T00:00:07.037Z'
        numberOfData: 1
        locationName: 'Tsawwassen - Duke Point'

    queryMetadata: [1x1 struct]
        rowLimit: 100000
        begin: '2016-09-01T00:00:00.000Z'
        end: '2016-09-01T00:00:10.000Z'
        deviceCategoryCode: 'TSG'
        locationCode: 'TWDP'

```

Property	Type	Description
data	struct	
data[].rawData	string	A raw data string representing a sample at a specific time.
data[].sampleTime	datetime string	The recorded time of the observation. <ul style="list-style-type: none"> Date Time format: yyyy-MM-dd'T'HH:mm:ss.SSS'Z'. datetime is represented in Coordinated Universal Time (UTC).
metadata	struct	A structure containing metadata.
metadata.dataMetadata	struct	A structure containing metadata about the data.
metadata.dataMetadata.firstTimestamp	datetime string	The date time of the first record in yyyy-MM-ddTHH:mm:ss.SSSZ format.

metadata.dataMetadata.hasMoreDataInRange	int	Whether more data are available within the range of the request. For example if 1,000 records are returned (because the rowLimit=1000), but there should be 200,000 rows as defined by the filter criteria. <ul style="list-style-type: none"> • 0 = False • 1 = True
metadata.dataMetadata.lastTimestamp	datetime string	The date time of the last record in yyyy-MM-ddTHH:mm:ss.SSSZ format.
metadata.dataMetadata.numberOfData	int	The number of samples in the data list.
metadata.dataMetadata.locationName	string	The full name of the location.
metadata.queryMetadata	struct	A structure containing metadata about the data request (query).
metadata.queryMetadata.rowLimit	int	The requested row limit per sensor.
metadata.queryMetadata.begin	datetime string	The begin date of the request in yyyy-MM-ddTHH:mm:ss.SSSZ format.
metadata.queryMetadata.end	datetime string	The end date of the request in yyyy-MM-ddTHH:mm:ss.SSSZ format.
metadata.queryMetadata.deviceCategoryCode	string	The Device Category Code of the request. See Available Device Categories for further details.
metadata.queryMetadata.locationCode	string	The Location Code of the request. See Available Locations for further details.

Example - Print the last raw instrument reading from the CTD at Barkley Canyon Axis

```
o = onc('YOUR_TOKEN_HERE');
payload = o.getDirectRawByStation(struct('locationCode','BACAX','deviceCategoryCode','CTD'),1);
disp(payload);
```

Example - Print the last raw instrument reading from AIS Reiever at the Underwater Listening Station at IONA

```
o = onc('YOUR_TOKEN_HERE');
payload = o.getDirectRawByStation(struct('locationCode','IONA','deviceCategoryCode','AISRECEIVER'),1);
disp(payload);
```

Example - Print 1 hour of raw CTD readings from Barkley Canyon Axis

```
o = onc('YOUR_TOKEN_HERE');
payload = o.getDirectRawByStation(struct('locationCode','BACAX','deviceCategoryCode','CTD','begin','2017-05-23T00:00:00.000Z','end','2017-05-23T01:00:00.000Z'));
disp(payload);
```

```
getDirectRawByLocation(filters)
```


Returns raw data from an instrument in the payload, in JSON format that meet the criteria defined by the input filters.

Parameter	Type	Description	Example
Required			
filter	struct or cell array	<p>A comma separated list of filter criteria parameters, used to request data and metadata from the rawdata endpoint.</p> <ul style="list-style-type: none"> A list of filter/value pairs in the following format struct('<filter_1>','<value_1>','<filter_2>','<value_2>',...)' or {'<filter_1>','<value_1>','<filter_2>','<value_2>',...}. Filters must include locationCode and deviceCategoryCode. Filters can optionally include dateFrom and dateTo (a date range). Filters can optionally specify "rowLimit", the maximum number of rows of data that will be returned for each sensor. <ul style="list-style-type: none"> The maximum is 100,000. If omitted or a higher or invalid limit is specified, the default of 100,000 is used. See locations Discovery Service and deviceCategories Discovery Service for more information on filter usage. 	<pre>struct ('locationCode','TWDP', ... 'deviceCategoryCode','TSG', ... 'dateFrom','2016-09-01T00: 00:00.000Z', ... 'dateTo','2016-09-01T08:00: 00.000Z') or {'locationCode','TWDP', ... 'deviceCategoryCode','TSG', ... 'dateFrom','2016-09-01T00: 00:00.000Z', ... 'dateTo','2016-09-01T08:00: 00.000Z'}</pre>

Returns

A structure that contains sensor data matching the filter criteria defined in the input parameters, metadata and service metadata.

```
data: [1x1 struct]
metadata: [1x1 struct]

next: [1x1 struct]
```

Property	Type	Description
data	struct	
data.readings[]	string	A raw data string representing a sample at a specific time.
data.times[]	datetime string	<p>The recorded time of the observation.</p> <ul style="list-style-type: none"> Date Time format: <code>yyyy-MM-dd'T'HH:mm:ss.SSS'Z'</code>. datetime is represented in Coordinated Universal Time (UTC).
metadata.locationName	string	The full name of the location.
next	struct	A structure containing metadata.
next.parameters	struct	A structure containing metadata about the data.
next.url	string	A string containing the query url.
next.parameters.rowLimit	int	The requested row limit per sensor.
next.parameters.begin	datetime string	The begin date of the request in yyyy-MM-ddTHH:mm:ss.SSSZ format.
next.parameters.end	datetime string	The end date of the request in yyyy-MM-ddTHH:mm:ss.SSSZ format.
next.parameters.deviceCategoryCode	string	The Device Category Code of the request. See Available Device Categories for further details.
next.parameters.locationCode	string	The Location Code of the request. See Available Locations for further details.

Example - Print the last raw instrument reading from the CTD at Barkley Canyon Axis

```
o = onc('YOUR_TOKEN_HERE');

payload = o.getDirectRawByLocation(struct('locationCode','BACAX','deviceCategoryCode','CTD','rowLimit',1));

disp(payload);
```

Example - Print the last raw instrument reading from AIS Reiever at the Underwater Listening Station at IONA

```
o = onc('YOUR_TOKEN_HERE');

payload = o.getDirectRawByLocation(struct('locationCode','IONA','deviceCategoryCode','AISRECEIVER','rowLimit',1));

disp(payload);
```

Example - Print 1 hour of raw CTD readings from Barkley Canyon Axis

```
o = onc('YOUR_TOKEN_HERE');

payload = o.getDirectRawByLocation(struct('locationCode','BACAX','deviceCategoryCode','CTD','dateFrom','2017-05-23T00:00:00.000Z','dateTo','2017-05-23T01:00:00.000Z'));

disp(payload);
```

```
getDirectRawByDevice(filters)
```

Returns raw data from an instrument in the payload, in JSON format that meet the criteria defined by the input filters.

Parameter	Type	Description	Example
Required			
filter	struct or cell array	<p>A comma separated list of filter criteria parameters, used to request data and metadata from the rawdata endpoint.</p> <ul style="list-style-type: none">• A list of filter/value pairs in the following format struct('<filter_1>','<value_1>','<filter_2>','<value_2>',...) or {'<filter_1>','<value_1>','<filter_2>','<value_2>',...}.• Filters must include deviceCode.• Filters can optionally include dateFrom and dateTo (a date range).• Filters can optionally specify "rowLimit", the maximum number of rows of data that will be returned for each sensor.<ul style="list-style-type: none">• The maximum is 100,000. If omitted or a higher or invalid limit is specified, the default of 100,000 is used.• See locations Discovery Service and deviceCategories Discovery Service and devices Discovery Service for more information on filter usage.	<pre>struct ('deviceCode','ICLISTENHF1 293', ... 'dateFrom','2016-09-01T00: 00:00.000Z', ... 'dateTo','2016-09-01T08:00: 00.000Z') or {'deviceCode','ICLISTENHF1 293', ... 'dateFrom','2016-09-01T00: 00:00.000Z', ... 'dateTo','2016-09-01T08:00: 00.000Z'}</pre>

Returns

A structure that contains sensor data matching the filter criteria defined in the input parameters, metadata and service metadata.

```
data: [1x1 struct]
next: [1x1 struct]
```

Property	Type	Description
data	struct	
data.readings[]	string	A raw data string representing a sample at a specific time.
data.times[]	datetime string	The recorded time of the observation. <ul style="list-style-type: none">• Date Time format: <code>yyyy-MM-ddT'HH:mm:ss.SSS'Z'</code>.• datetime is represented in Coordinated Universal Time (UTC).
next	struct	A structure containing metadata.
next.parameters	struct	A structure containing metadata about the data.
next.url	string	A string containing the query url.
next.parameters.rowLimit	int	The requested row limit per sensor.
next.parameters.begin	datetime string	The begin date of the request in <code>yyyy-MM-ddTHH:mm:ss.SSSZ</code> format.
next.parameters.end	datetime string	The end date of the request in <code>yyyy-MM-ddTHH:mm:ss.SSSZ</code> format.
next.parameters.deviceCode	string	The Device Code of the request. See Available Device Categories for further details.

Example - Print the last raw instrument reading from the CTD at Barkley Canyon Axis

```
o = onc('YOUR_TOKEN_HERE');
payload = o.getDirectRawByDevice(struct('deviceCode','ICLISTENHF1293','rowLimit',1));
disp(payload);
```

Example - Print the last raw instrument reading from AIS Reiever at the Underwater Listening Station at IONA

```
o = onc('YOUR_TOKEN_HERE');
payload = o.getDirectRawByDevice(struct('deviceCode','ICLISTENHF1293','rowLimit',1));
disp(payload);
```

Example - Print 1 hour of raw CTD readings from Barkley Canyon Axis

```
o = onc('YOUR_TOKEN_HERE');
payload = o.getDirectRawByDevice(struct('deviceCode','BACAX','dateFrom','2017-05-23T00:00:00.000Z','dateTo','2017-05-23T01:00:00.000Z'));
disp(payload);
```

```
getDirectFiles( filters )
```

Returns data files from an instrument in the payload, in JSON format that meet the criteria defined by the input filters.

Parameter	Type	Description	Example
Required			
filters	struct or cell array	<p>A comma separated list of filter criteria parameters, used to request data and metadata from the rawdata endpoint.</p> <ul style="list-style-type: none"> A list of filter/value pairs in the following format struct('<filter_1>','<value_1>','<filter_2>','<value_2>',...) or {'<filter_1>','<value_1>','<filter_2>','<value_2>',...}. Filters must include locationCode. Filters can optionally include dateFrom and dateTo. See locations Discovery Service for more information on filter usage. 	<pre>struct ('locationCode','TWDP', ... 'deviceCategoryCode','TSG', ... 'dateFrom','2016-09-01T00:00:00.000Z', ... 'dateTo','2016-09-01T08:00:00.000Z') or {'locationCode','TWDP', ... 'deviceCategoryCode','TSG', ... 'dateFrom','2016-09-01T00:00:00.000Z', ... 'dateTo','2016-09-01T08:00:00.000Z'}</pre>

Returns

A structure that contains sensor data matching the filter criteria defined in the input parameters, metadata and service metadata.

```
data: [1x1 struct]
    rawData: '20160901-00:00:07.037, 12.9162, 3.44294, 28.9096'
    sampleTime: '2016-09-01T00:00:07.037Z'

metadata: [1x1 struct]
    dataMetadata: [1x1 struct]
        firstTimestamp: '2016-09-01T00:00:07.037Z'
        hasMoreDataInRange: 0
        lastTimestamp: '2016-09-01T00:00:07.037Z'
        numberOfData: 1
        locationName: 'Tsawwassen - Duke Point'

    queryMetadata: [1x1 struct]
        rowLimit: 100000
        begin: '2016-09-01T00:00:00.000Z'
        end: '2016-09-01T00:00:10.000Z'
        deviceCategoryCode: 'TSG'
        locationCode: 'TWDP'
```

Property	Type	Description
data	struct	

Example - Print the last raw instrument reading from the CTD at Barkley Canyon Axis

```
o = onc('YOUR_TOKEN_HERE');

payload = o.getDirectFiles(struct('locationCode','BACAX','deviceCategoryCode','CTD'),1);

disp(payload);
```

Example - Print the last raw instrument reading from AIS Reviewer at the Underwater Listening Station at IONA

```
o = onc('YOUR_TOKEN_HERE');  
  
payload = o.getDirectFiles(struct('locationCode','IONA','deviceCategoryCode','AISRECEIVER'),1);  
  
disp(payload);
```

Example - Print 1 hour of raw CTD readings from Barkley Canyon Axis

```
o = onc('YOUR_TOKEN_HERE');  
  
payload = o.getDirectFiles(struct('locationCode','BACAX','deviceCategoryCode','CTD','dateFrom','2017-05-23T00:00:00.000Z','dateTo','2017-05-23T01:00:00.000Z'));  
  
disp(payload);
```



Please report all issues with the web services, documentation, samples and client libraries to the [Oceans 2.0 Help Centre](#)