

dataProductDelivery Service

- [Description](#)
 - [URL](#)
- [request](#)
 - [Parameters](#)
 - [Determining Search Type \(Valid combinations\)](#)
 - [Response](#)
 - [Success for files from Archive \(HTTP 200\)](#)
 - [Success for generated data products \(HTTP 200\)](#)
 - [Bad Request \(HTTP 400\)](#)
 - [Examples](#)
- [status](#)
 - [Parameters](#)
 - [Response](#)
 - [Success \(HTTP 200\)](#)
 - [Bad Request \(HTTP 400\)](#)
 - [Example](#)
- [run](#)
 - [Parameters](#)
 - [Response](#)
 - [Success \(HTTP 200\)](#)
 - [Accepted \(HTTP 202\)](#)
 - [Unauthorized \(HTTP 401\)](#)
 - [Bad Request \(HTTP 400\)](#)
 - [Example](#)
- [download](#)
 - [Parameters](#)
 - [Response](#)
 - [Success](#)
 - [Bad Request \(HTTP 400\)](#)
 - [Response Codes](#)
 - [Examples](#)
- [Cancel](#)
 - [Parameters](#)
 - [Response](#)
 - [Success \(HTTP 200\)](#)
 - [Unauthorized \(HTTP 401\)](#)
 - [Bad Request \(HTTP 400\)](#)
 - [Example](#)
- [Restart](#)
 - [Parameters](#)
 - [Response](#)
 - [Success \(HTTP 200\)](#)
 - [Unauthorized \(HTTP 401\)](#)
 - [Bad Request \(HTTP 400\)](#)
 - [Example](#)
 - [Code Examples](#)

Description

The API **dataProductDelivery** service provides the ability to request, run and download Data Products for the following:

1. Locations & Device Categories
2. Locations, Device Categories & Properties
3. Locations & Properties
4. Devices
5. Devices & Properties

Downloading a data product is done with a three-step process:

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

Validates request parameters, creates a new request definition and provides performance estimates

- Use filters to supply the Who, What, Where, When and How of the data product request
 - [Who](#): Token
 - [What](#): Device Category and/or Property
 - [Where](#): Location or Device
 - [When](#): dateFrom and dateTo date/time
 - [How](#): Data Product, extension and Data Product Options

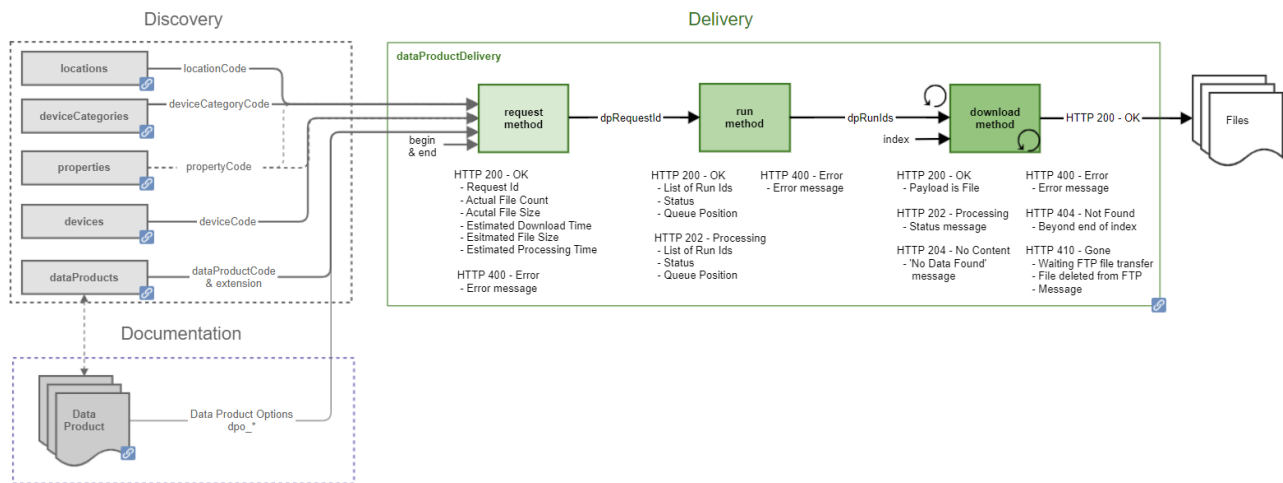
2. Run the data product request using the **run** method

Starts the data product generation process by adding the request to the Task Queue

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

Attempts to download a file at an index. If the file at that index is not ready for download at the time of the download request, the service returns information about the status of the process. The client application must continue to request the download until the file is ready, or an error is encountered.

You can optionally check the status of a data product request (even an already finished one) with the **status** method.



(internal source for the above diagram: <https://internal.oceannetworks.ca/display/ONCData/Services+Workflow+-+dataProductDelivery>)

URL

<https://data.oceannetworks.ca/api/dataProductDelivery>

Method	Description	Example
request	Create a request for a data product from a filter criteria parameters.	method=request
status	Retrieve the processing status of a request.	method=status
run	Run a data product created by the request method.	method=run
cancel	cancels a running data product	method=cancel
restart	restarts a cancelled data product	method=restart
download	Download one of the data product result files by index number.	method=down Load

request

The request method creates a search and returns information regarding the number of files, file sized, compressed file sized, and estimated download times as well as a request id that can be used to generate the data product using the run method.

Parameters

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
dataProductCode	string	Request a Data Product for a specific Data Product Code . - Data Product Code must be valid. - Specific Data Product Codes can be obtained programmatically from the dataProducts service or interactively from the Available Data Products page.	dataProductCode=TSSD
extension	string	Request a Data Product for a specific File Format Extension . - Extension must be valid and match exactly, including case. - Extensions available for specific Data Product Codes can be obtained programmatically from the dataProducts service or interactively from the Available Data Products page.	extension=csv
dateFrom	datetime	Request a Data Product starting at a specific date/time. Accepted DateTime formats: - yyyy-MM-dd'T'HH:mm:ss.SSS'Z' (ISO 8601 Extended) - yyyy-MM-dd (ISO 8601 Extended) - PnYnMnDTnHnMnS (ISO 8601 Duration) - DateTime is represented in Coordinated Universal Time (UTC). - ISO 8601 Extended format without a time will be assumed to mean midnight (T00:00:000.000Z). - Queries with both dateFrom and dateTo in the ISO 8601 Duration format will not be accepted.	dateFrom=2016-07-25T00:00:00.000Z dateFrom=2010-07-27 dateFrom=-P1DT1H Previous 1 day and 1 hour, relative to the dateTo. Note the '-' before the P.
dateTo	datetime	Request a Data Product ending at a specific date/time. Accepted DateTime formats: - yyyy-MM-dd'T'HH:mm:ss.SSS'Z' (ISO 8601 Extended) - yyyy-MM-dd (ISO 8601 Extended) - PnYnMnDTnHnMnS (ISO 8601 Duration) - DateTime is represented in Coordinated Universal Time (UTC). - ISO 8601 Extended format without a time will be assumed to mean midnight (T00:00:000.000Z). - Queries with both dateFrom and dateTo in the ISO 8601 Duration format will not be accepted.	dateTo=2016-07-29T00:00:00.000Z dateTo=2016-08-01 dateTo=PT12H30M Next 12 hours and 30 minutes, relative to the dateFrom.
Conditionally Required			
locationCode	string	Request a Data Product from a specific Location . - Location Code must be valid. - Specific Location Codes can be obtained programmatically from the locations service or interactively from the Available Locations page.	locationCode=BACAX
deviceCategoryCode	string	Request a Data Product from devices belonging to a specific Device Category . - Device Category Code must be valid. - Specific Device Category Codes can be obtained programmatically from the deviceCategories service or interactively from the Available Device Categories page.	deviceCategoryCode=ADCP2MHZ
deviceCode	string	Request a Data Product from a specific Device . - Device Code must be valid. - Specific Device Codes can be obtained programmatically from the devices service or interactively from the Available Devices page.	deviceCode=AandOpt0581
propertyCode	string	Request a Data Product for a specific Property . - Property Code must be valid. - Specific Property Codes can be obtained programmatically from the properties service or interactively from the Available Properties page.	property=pressure
Data Product Options	string	Each Data Product Extension may have required data product options that determine how to process or package the data. - Data Product Options include and are not limited to: - Quality Control - Resampling - Data Gaps - Hydrophone Data Diversion Mode - Data Product Options and usage information for each Data Product, can be found on the Data Product Options page.	dpo_qualityControl=1 dpo_resample=average dpo_average=600

Determining Search Type (Valid combinations)

Search Type Equivalent	Parameters Given
Instrument by Location (Device Level)	locationCode, deviceCategoryCode
Instrument by Location (Sensor Level)	locationCode, deviceCategoryCode, propertyCode
Instrument by Category (Device Level)	deviceCode
Instrument by Category (Sensor Level)	deviceCode, propertyCode
Variable by Location	locationCode, propertyCode

Response

Detailed information about the number of files, file size, compressed file size, and download time estimates are returned if the files are in archive. If the data product needs to be generated, a dpRequestId, status and filesize and processing time estimates are returned.

Success for files from Archive (HTTP 200)

```
{
  "compressedFileSize": 12563408,
  "downloadTimes": {
    "10Mbps": 7.076623,
    "50Mbps": 1.4153247,
    "150Mbps": 0.47177488
  },
  "dpRequestId": 1852695,
  "fileSize": 70766230,
  "numFiles": 4
}
```

Property	Type	Description	Example
dpRequestId	integer	Returns a numerical request id to be passed into the "run" method (searchHdrId).	"dpRequestId":1852695
numFiles	integer	Returns the number of files that meet the criteria of the search	"numFiles":4
fileSize	Long	The sum of all file sizes (in bytes)	"fileSize":70766230
compressedFileSize	Long	The sum of all file size (in bytes) after compression	"compressedFileSize":12563408
downloadTimes	String:' Float	A set of three common download speeds and the time (in seconds) it would take to download the files for each speed	"downloadTimes":{ "10Mbps":7.076623, "50Mbps":1.4153247, "150Mbps":0.47177488 }

Success for generated data products (HTTP 200)

```
{
  "disclaimer": "Software Engineering is implementing estimates of processing times and file sizes for data requests. These are extremely rough to begin with, but bear with us. We expect these estimates will gradually improve.",
  "dpRequestId": 1908432,
  "estimatedFileSize": "11940760",
  "estimatedProcessingTime": "21"
}
```

Property	Type	Description	Example
dpRequestId	integer	Returns a numerical request id to be passed into the "run" method (SearchHdrId)	"dpRequestId":1908432
estimatedFileSize	string	The estimated file size (in bytes)	"estimatedFileSize": "11940760" or "estimatedFileSize": "no estimated file size available."
estimatedProcessingTime	string	The estimated time (in seconds) it will take to process the data product	"estimatedProcessingTime": "21" or "estimatedProcessingTime": "No processing time estimates found for this type of data."

disclaimer	string	A statement saying that these estimates are rough	"disclaimer":"Software Engineering is implementing estimates of processing times and file sizes for data requests. These are extremely rough to begin with, but bear with us. We expect these estimates will gradually improve."
------------	--------	---	--

Bad Request (HTTP 400)

errorCode	errorMessage	Description
23	Invalid Time Range, Start Time is greater than End Time or start time is not provided	Occurs when the dateTo is before the dateFrom date/time. The name of both of the datetime filters will be included in the "parameter" property
25	Invalid Time Range, Start Time is in the future.	Occurs when the end is before the begin date/time. The names of both of the datetime filters are included in the parameter property.
33	No data found for data search with given parameters	Occurs when there is no data available for given parameters
33	No site devices found for this device in the given time range.	Occurs when there is no site device available for devices obtained from parameters in the given time range
71	Supplied deviceCategoryCode does not have corresponding dataProductCode and extension.	Occurs when deviceCategoryCode does not have corresponding dataProductCode and extension dataProducts service can be used to get the list of valid dataProductCodes and extensions available for the given deviceCategoryCode
71	Supplied deviceCode does not have corresponding dataProductCode and extension.	Occurs when deviceCode does not have corresponding dataProductCode and extension dataProducts service can be used to get the list of valid dataProductCodes and extensions available for the given deviceCode
71	Supplied propertyCode does not have corresponding dataProductCode and extension.	Occurs when propertyCode does not have corresponding dataProductCode and extension dataProducts service can be used to get the list of valid dataProductCodes and extensions available for the given propertyCode
71	Data restricted by [ORGANIZATION_NAME]. To access, get written permission via [ONC_CONTACT_INFO]. The latest date available is [END_DATE]	Occurs when the data in the requested time period has restricted access
127	Supplied property does not have corresponding dataProductCode and extension	Occurs when propertyCode does not have corresponding dataProductCode and extension dataProducts service can be used to get the list of valid dataProductCodes and extensions available for the given propertyCode
127	Supplied deviceCategoryCode does not have corresponding dataProductCode and extension	Occurs when deviceCategoryCode does not have corresponding dataProductCode and extension dataProducts service can be used to get the list of valid dataProductCodes and extensions available for the given deviceCategoryCode
127	End parameter value must be greater than begin parameter value	Occurs when dateTo date equals or is less than dateFrom date
127	Invalid parameter value	Occurs when an invalid code is used in the filter. Most filters require an exact match, otherwise this error will occur. The name of the filter parameter is included in the parameter property.

128	Missing parameter	Occurs when multiple parameters are needed, but not all are present. Occurs when dateFrom is used without dateTo or vice versa. The names of the required filter parameters, separated by /, are included in the parameter property.
128	Requested type can not be determined	Occurs when the one of the 5 possible parameter combinations is not specified. A list of the valid parameter combinations are listed in the parameter. "possible parameter combinations are: locationCode, deviceCategoryCode; locationCode, deviceCategoryCode, propertyCode; locationCode, propertyCode; deviceCode; deviceCode, propertyCode"
129	Invalid parameter name	Occurs when an unsupported filter parameter is used. The name of the filter parameter will be included in the "parameter" property
39	dataProductCode and extension not a valid dataProductFormat	Occurs when combination of dataProductCode and extension is not a valid dataProductFormat

Examples

- Request a 'Log File' data product in 'txt' format, using **Data Product Code** 'LF' and **Extension** 'txt' for **Device Category Code** 'ADCP2MHZ' at the **Location Code** 'BACAX', **Between** 25-28 July 2016.
[https://data.oceannetworks.ca/api/dataProductDelivery?method=request&token=\[YOUR_TOKEN_HERE\]&locationCode=BACAX&deviceCategoryCode=ADCP2MHZ&dataProductCode=LF&extension=txt&dateFrom=2016-07-25T00:00:00.000Z&dateTo=2016-07-29T00:00:00.000Z](https://data.oceannetworks.ca/api/dataProductDelivery?method=request&token=[YOUR_TOKEN_HERE]&locationCode=BACAX&deviceCategoryCode=ADCP2MHZ&dataProductCode=LF&extension=txt&dateFrom=2016-07-25T00:00:00.000Z&dateTo=2016-07-29T00:00:00.000Z)
- Request a 'Time Series Scalar Data' data product in 'csv' format, using **Data Product Code** 'TSSD' and **Extension** 'csv' for **Property Code** 'salinity' at 'Barkley Canyon Axis (Pod 1) location using **Location Code** 'BACAX', **Between** 1-12 July 2017.
[https://data.oceannetworks.ca/api/dataProductDelivery?method=request&token=\[YOUR_TOKEN_HERE\]&locationCode=BACAX&propertyCode=salinity&dataProductCode=TSSD&extension=csv&dateFrom=2017-01-01T00:00:00.000Z&dateTo=2017-01-13T00:00:00.000Z&dpo_qualityControl=1&dpo_resample=none&dpo_dataGaps=0](https://data.oceannetworks.ca/api/dataProductDelivery?method=request&token=[YOUR_TOKEN_HERE]&locationCode=BACAX&propertyCode=salinity&dataProductCode=TSSD&extension=csv&dateFrom=2017-01-01T00:00:00.000Z&dateTo=2017-01-13T00:00:00.000Z&dpo_qualityControl=1&dpo_resample=none&dpo_dataGaps=0)

status

The status method returns data about the status of the request.

Parameters

Parameter	Type	Description	Example
Required			
token	string	All Web Services require a token. This can be generated at https://data.oceannetworks.ca/Profile . Click on the "Web Services" tab and click "Generate Token"	token=YOUR_TOKEN_HERE
dpRequestId	integer	A dpRequestId returned from the request method (searchHdrId).	dpRequestId=1852695

Response

Success (HTTP 200)

```
{
  "cartId": null,
  "cartStatus": 0,
  "description": "",
  "fixed_dates": false,
  "modifyBy": 49860,
  "modifyDate": "2018-02-02T19:22:19.555Z",
```

```

"name": "",
"newSearches": [
  {
    "dataProductFormat": {
      "archiveFileTypeId": 12,
      "dataProductFormatId": 2,
      "dataProductFormatName": null,
      "dataProductId": 1,
      "extension": {
        "archiveFileExtensionId": 3,
        "description": "CSV file",
        "extension": "csv",
        "mimeType": "text/plain"
      },
      "mode": null,
      "modifyBy": 2685,
      "modifyDate": "2009-11-12T10:54:08.000Z"
    },
    "dateCreated": "2018-02-02T19:22:19.558Z",
    "dateFrom": "2017-01-01T00:00:00.000Z",
    "dateTo": "2017-01-13T00:00:00.000Z",
    "deviceCategoryName": "CTD",
    "deviceId": null,
    "deviceName": null,
    "formatId": null,
    "interopId": null,
    "interopIdentifier": null,
    "interopSubIdentifier": null,
    "itemId": null,
    "maxDepth": null,
    "maxLat": null,
    "maxLon": null,
    "minDepth": null,
    "minLat": null,
    "minLon": null,
    "notification": null,
    "parameter": "{\"35\":\"0\",\"38\":\"1\",\"40\":\"41\",\"0\"}",
    "productId": null,
    "results": null,
    "searchDtIs": [
      {
        "deviceId": 23842,
        "deviceName": "Sea-Bird SeaCAT SBE19plus V2 6813",
        "sensorId": 16702,
        "sensorName": "Practical Salinity"
      }
    ],
    "searchId": 5968038,
    "searchNodeId": 15,
    "searchTypeId": 8,
    "sensorId": null,
    "sensorName": "Salinity",
    "siteDeviceId": null,
    "slow": false,
    "subsamplePeriod": null,
    "subsampleType": null,
    "taskHistoryId": 18553457,
    "taskStatus": "Completed",
    "venusSearchParameter": {
      "dataFormatId": 0,
      "extraParameters": "{\"35\":\"0\",\"38\":\"1\",\"40\":\"41\",\"0\"}",
      "locationId": 0,
      "regionId": 0,
      "siteId": 0,
      "studyAreaId": 0,
      "waterPropertyId": 0
    }
  }
],
"runningSearches": {
},
"searchHdrId": 2609760,
"searchHdrStatus": "COMPLETED",
"searchSourceId": 5
}

```

Property	Type	Description	Example
Property	Type	Description	Example
cartId	string	Deprecated. The ID of the cart/search header. Can be null.	"cartId": null
cartStatus	integer	The status of the cart/search header as a number. What those numbers correspond to are below: 0: Closed 1: Open	"cartStatus": 1
description	string	A description of the search header. Can be empty string.	"description": "test"
fixed_dates	boolean	Whether this search's dates are fixed through the delivery API.	"fixed_dates": false
modifyBy	integer	The ID of the DMAS user who last modified the search header.	"modifyBy": 19372
modifyDate	string	The date of the last time the search header was modified. This is in the form of a string in ISO8601 (extended) format, yyyy-MM-dd'T'HH:mm:ss.SSS'Z'.	"modifyDate": "2017-05-09T21:43:29.000Z"
name	string	The name of the search header. Can be empty string.	"name": {}
searchHdrId	integer	The ID of the status results/search header.	"searchHdrId": 2007714
searchHdrStatus	string	The status of the search header.	"searchHdrStatus": "OPEN"
searchSourceId	integer	The search source as a number. What those numbers correspond to are below: 1: Venus page 2: DMAS page 3: Web service call 4: Multimedia search 5: Delivery API	"searchSourceId": 3
runningSearches	object	Supposedly a map that correlates numbers to currently running searches. Stays empty in practice.	"runningSearches": {}
newSearches	list	A list of new searches for that cart/search header.	
newSearches.dateCreated	string	The date the search was created. This is in the form of a string in ISO8601 (extended) format, yyyy-MM-dd'T'HH:mm:ss.SSS'Z'.	"dateCreated": "2017-05-09T21:43:29.000Z"
newSearches.dateFrom	string	The date of the start of the search range. This is in the form of a string in ISO8601 (extended) format, yyyy-MM-dd'T'HH:mm:ss.SSS'Z'.	"dateFrom": "2011-09-10T21:33:39.000Z"
newSearches.dateTo	string	The date of the end of the search range. This is in the form of a string in ISO8601 (extended) format, yyyy-MM-dd'T'HH:mm:ss.SSS'Z'.	"dateTo": "2017-01-13T00:00:00.000Z"
newSearches.deviceCategoryName	string	The name of the search's device category.	"deviceCategoryName": "BPR"
newSearches.deviceId	integer	The ID of the search's device.	"deviceId": 10701
newSearches.deviceName	string	The name of the search's device.	"deviceName": "NRCan Bottom Pressure Recorder 06"
newSearches.formatId	integer	Supposedly the search's data product format ID. Null in practice. May be completely replaced by newSearches.dataProductFormat in the future.	"formatId": null
newSearches.interopId	integer	The ID of the Interop search associated with this search (if any). Can be null.	"interopId": 650
newSearches.interopIdentifier	string	The identifier of the Interop search associated with this search (if any). Can be null.	"interopIdentifier": "linep"

newSearches.interopSubIdentifier	string	The sub-identifier of the Interop search associated with this search (if any). Can be null.	"interopSubIdentifier": "AMON"
newSearches.itemId	string	Supposedly a unique ID generated by the UI that corresponds to the search line of this search. Can be null.	"itemId": null
newSearches.maxDepth	double	The maximum depth of this search. Can be null.	"maxDepth": null
newSearches.maxLat	double	The maximum latitude of this search. Can be null.	"maxLat": null
newSearches.maxLon	double	The maximum longitude of this search. Can be null.	"maxLon": null
newSearches.minDepth	double	The minimum depth of this search. Can be null.	"minDepth": null
newSearches.minLat	double	The minimum latitude of this search. Can be null.	"minLat": null
newSearches.minLon	double	The minimum longitude of this search. Can be null.	"minLon": null
newSearches.notification	string	A notification associated with this search. Can be null.	"notification": null
newSearches.parameter	string	Additional parameters and their values for the search as coded numbers in a stringified object. These correspond to parameters like "dpo_resample".	"parameter": "{ \"35\": \"1\", \"38\": \"1\", \"40\": [\"39\", \"3600\"] }"
newSearches.productId	integer	The product ID of the search. Null in practice.	"productId": null
newSearches.results	object	A search results summary for the search. Can be null.	"results": null
newSearches.searchId	integer	The ID of the search.	"searchId": 4433200
newSearches.searchNodeId	integer	The ID of the search node associated with the search.	"searchNodeId": 15
newSearches.searchTypeId	integer	The search type as a number. What those numbers correspond to are below: 0: Venus 1: General 5: Station/location 7: Web service 8: Primary sensor	"searchTypeId": 1
newSearches.sensorId	integer	The ID of the search's sensor.	"sensorId": 9061
newSearches.sensorName	string	The name of the search's sensor.	"sensorName": "Housing Temperature"
newSearches.siteDeviceId	integer	The ID of the siteDevice associated with the search.	"siteDeviceId": 1194492
newSearches.slow	boolean	Whether this search is slow. Set to false in practice.	"slow": false
newSearches.subsamplePeriod	integer	The subsampling period in seconds. Analogous to part of the "Processing" section of Data Product Options in Data Search.	"subsamplePeriod": 3600

newSearches.subsampleType	integer	The type of subsampling the search uses as a number. What those numbers correspond to are below: 1: Average 2: Decimated 3: Min/Max	"subsampleType": 1
newSearches.taskHistoryId	integer	The ID of the task history associated with the search. Can be null, especially when the search has not been started or queued yet.	"taskHistoryId": 18553457
newSearches.taskStatus	string	The status of the task history associated with the search. Can be null, especially when the search has not been started or queued yet.	"taskStatus": "Completed"
newSearches.dataProductFormat	object	The data product format of the search.	
- newSearches.dataProductFormat.archiveFileTypeId	integer	The ID of the archive file type of the data product format.	"archiveFileTypeId": 43
- newSearches.dataProductFormat.dataProductId	integer	The ID of the data product format.	"dataProductId": 145
- newSearches.dataProductFormat.dataProductFormatName	string	The name of the data product format. Can be null.	"dataProductFormatName": null
- newSearches.dataProductFormat.dataProductId	integer	The ID of the data product associated with that data product format.	"dataProductId": 114
- newSearches.dataProductFormat.mode	string	The file mode of the data product format. Can be null.	"mode": "-ODV"
- newSearches.dataProductFormat.modifyBy	integer	The ID of the DMAS user who last modified the data product format.	"modifyBy": 29942
- newSearches.dataProductFormat.modifyDate	string	The date of the last time the data product format was modified. This is in the form of a string in ISO8601 (extended) format, yyyy-MM-dd'T'HH:mm:ss.SSS'Z'.	"modifyDate": "2015-04-01T16:36:24.000Z"
- newSearches.dataProductFormat.extension	object	More details about the extension of the data product format.	
- newSearches.dataProductFormat.extension.archiveFileExtensionId	integer	The ID of the extension.	"archiveFileExtensionId": 33
- newSearches.dataProductFormat.extension.description	string	A description of the extension.	"description": "RUV file"
- newSearches.dataProductFormat.extension.extension	string	The extension itself (without the .).	"extension": "ruv"

newSearches.dataProductFormat.extension.mimeType	string	The MIME type of the file format corresponding to that extension.	"mimeType": "application/octet-stream"
newSearches.searchDtIs	list	A list of sets of details for this search. Can be null.	
- newSearches.searchDtIs.deviceId	integer	The device ID for this set of search details.	"deviceId": 23842
- newSearches.searchDtIs.deviceName	string	The device name for this set of search details.	"deviceName": "Sea-Bird SeaCAT SBE19plus V2 6813"
- newSearches.searchDtIs.sensorId	integer	The sensor ID for this set of search details.	"sensorId": 16702
- newSearches.searchDtIs.sensorName	string	The sensor name for this set of search details.	"sensorName": "Practical Salinity"
newSearches.venusSearchParameter	object	The Venus search parameter of the search. Not all details may be filled in well.	
- newSearches.venusSearchParameter.dataFormatId	integer	The data format ID of the Venus search parameter.	"dataFormatId": 0
- newSearches.venusSearchParameter.extraParameters	string	Additional parameters and their values for the Venus search as coded numbers in a stringified object. These correspond to parameters like "dpo_resample".	"extraParameters": "{\\"35\\":\\"1\\",\\"38\\":\\"1\\",\\"40\\":[\\"39\\",\\"3600\\"]}"
- newSearches.venusSearchParameter.locationId	integer	The location ID of the Venus search parameter.	"locationId": 0
- newSearches.venusSearchParameter.regionId	integer	The region ID of the Venus search parameter.	"regionId": 0
- newSearches.venusSearchParameter.siteId	integer	The site ID of the Venus search parameter.	"siteId": 1000730
- newSearches.venusSearchParameter.studyAreaId	integer	The study area ID of the Venus search parameter.	"studyAreaId": 0
- newSearches.venusSearchParameter.waterPropertyId	integer	The water property ID of the Venus search parameter.	"waterPropertyId": 0

Bad Request (HTTP 400)

errorCode	errorMessage	Description
127	Invalid parameter value	Occurs when an invalid code is used in the query. The name of the filter parameter will be included in the "parameter" property
127	Invalid search header ID.	Occurs when an invalid Request ID is used in the dpRequestId paramters.

128	Missing parameter	Occurs when multiple parameters are needed, but not all are present. The names of the required filter parameters will be included in the "parameter" property separated by /
129	Invalid parameter name	Occurs when a filter parameter is in the query but is not supported. The name of the filter parameter will be included in the "parameter" property

Example

https://data.oceannetworks.ca/api/dataProductDelivery?method=status&token=YOUR_TOKEN_HERE&dpRequestId=2007707

run

The run method runs the data product created by a call to the request method.

Parameters

Parameter	Type	Description	Example
Required			
token	string	All Web Services require a token. Once logged in at http://data.oceannetworks.ca/login , your token can be retrieved or generated at http://data.oceannetworks.ca/Profile . Click on the "Web Services" tab, then click "Generate Token".	token=YOUR_TOKEN_HERE
dpRequestId	integer	A dpRequestId returned from the request method (searchHdrId).	dpRequestId=1852695

Response

Success (HTTP 200)

HTTP status code 200 is returned when all of the runs in the data product delivery request have completed running.

```
[
  {
    "dpRunId": 4521400,
    "filecount": 5
    "status": "complete"
  }
]
```

The payload contains a list of runs, each containing the following properties:

Property	Type	Description	Example
dpRunId	integer	A Run Id that can be used to download the data product when processing is complete (searchId).	"dpRunId":4521400
fileCount	integer	The number of files available for download.	"fileCount":5
status	string	The current status of the data product request. Valid values are ("cancelled", "queued", "error", "running" and "complete").	"status": "complete"

Accepted (HTTP 202)

HTTP status code 202 is returned if all of the runs in the data product delivery request have not completed running.

```
[
  {
    "dpRunId": 4521400,
    "filecount": 5
    "status": "data product running"
  }
]
```

The payload contains a list of runs, each run can contain the following properties

Property	Type	Description	Example
dpRunId	integer	A Run Id that can be used to download the data product when processing is complete (searchId).	"dpRunId":4521400
fileCount	integer	The number of files available for download.	"fileCount":5
status	string	The current status of the data product request. Valid values are ("cancelled", "queued", "error", "running" and "complete").	"status": "complete"

Unauthorized (HTTP 401)

```
{
  "errors": [
    {
      "errorCode": 127,
      "errorMessage": "Invalid parameter value",
      "parameter": "token"
    }
  ]
}
```

errorCode	errorMessage	Description
127	Invalid parameter value	Occurs when an invalid token is used.
128	Missing parameter	Occurs when the token parameter is not included in the request

Bad Request (HTTP 400)

```
{
  "errors": [
    {
      "errorCode": 129,
      "errorMessage": "Invalid parameter name",
      "parameter": "RequestId"
    },
    {
      "errorCode": 128,
      "errorMessage": "Missing parameter",
      "parameter": "dpRequestId"
    }
  ]
}
```

errorCode	errorMessage	Description
-----------	--------------	-------------

127	Invalid parameter value	Occurs when an invalid code is used in the query. The name of the filter parameter will be included in the "parameter" property
127	Invalid search header ID	Occurs when an invalid Request ID is used in the dpRequestId parameters.
127	Search header has already run	Occurs when the same run request is made.
128	Missing parameter	Occurs when multiple parameters are needed, but not all are present. Occurs when begin is used without end. The names of the required filter parameters, separated by /, are included in the parameter property,
129	Invalid parameter name	Occurs when an unsupported filter parameter is used. The name of the filter parameter will be included in the "parameter" property

Example

[https://data.oceannetworks.ca/api/dataProductDelivery?method=run&token=\[YOUR_TOKEN_HERE\]&dpRequestId=\[YOUR_REQUEST_ID_HERE\]](https://data.oceannetworks.ca/api/dataProductDelivery?method=run&token=[YOUR_TOKEN_HERE]&dpRequestId=[YOUR_REQUEST_ID_HERE])

download

The download method downloads a file for the specified data product run request. The file to download is specified by index, with the first valid index being 1 and the last being the total number of files generated by the request.

Parameters

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
dpRunId	integer	The dpRunId returned from the run method (searchId).	dpRunId=5968038
index	integer / string	The index of the file to be downloaded, valid values are 1 to the number of result files. If the index is greater than the number of result files a response code of 204 is returned, indicating no file at that index. If index is string meta , metadata file will be downloaded	index=2 index=meta
deleteFile	Boolean	if deleteFile is set to false the file wont be deleted on the ONC servers after download. If deleteFile is true or not specified then the file will be deleted from the ONC servers after it is successfully downloaded (this is done to save working space on the servers)	deleteFile=false

Response

Success

HTTP status code 200 is returned in the HTTP header and the requested file is downloaded.

Bad Request (HTTP 400)

errorCode	errorMessage	Description
127	Invalid parameter value	Occurs when an invalid code is used in the query. The name of the filter parameter will be included in the "parameter" property
127	Index must be greater than 0	Occurs when index value is less than or equal to zero
127	Metadata file not valid for data product	Occurs when data product does not have metadata file

127	Invalid data product run ID	Occurs when invalid dpRunId is used in the query.
128	Missing parameter	Occurs when multiple parameters are needed, but not all are present. The names of the required filter parameters will be included in the "parameter" property separated by /
129	Invalid parameter name	Occurs when a filter parameter is in the query but is not supported. The name of the filter parameter will be included in the "parameter" property

Response Codes

Response Code	Description
200 - Ok	The file is included in the payload.
202 - Accepted	The data product and index are valid but the data product is still being processed. This status can be used to poll for data product completion, after first polling run or status for search completion. The payload includes the comment as displayed in the GUI.
204 - No Content	There is no result file for the data product at the index specified. This can be used to stop looping through result files when the actual number of result files isn't known.
400 - Bad Request	Missing required parameters.
	dpRunId is not a number.
	index is neither a positive number nor "meta".
	dpRunId is not a valid data product run Id.
	The data product selected by dpRunId was run by a different user.
	Index is "meta", but the data product does not include metadata
401 - Unauthorized	Token is invalid.
404 - Not Found	Index number is greater than the number of files.
410 - Gone	The result file selected by dpRunId and index is not in expected location and may have been deleted by the regular file clean up process or has been downloaded already and parameter deleteFile was true or not provided. This can also occur if there is a lag between the time that the data product generation is complete and when the file appears on the FTP server, however this scenario is usually caught as an http response code 202.
500 - Internal Server Error	This is an unhandled exception and should not occur. All 500 errors should be immediately reported and a JIRA ticket created at http://jira.neptune.uvic.ca/servicedesk/customer/portal/9/create/43 .

Examples

The following is an example of the return for a data product that is still being processed, note that the response code in this case is 202.

[https://data.oceannetworks.ca/api/dataProductDelivery?method=download&token=\[YOUR_TOKEN_HERE\]&dpRunId=\[YOUR_RUN_ID_HERE\]&index=\[YOUR_INDEX_#_HERE\]](https://data.oceannetworks.ca/api/dataProductDelivery?method=download&token=[YOUR_TOKEN_HERE]&dpRunId=[YOUR_RUN_ID_HERE]&index=[YOUR_INDEX_#_HERE])

```
{
  "message": "Transferring (BarkleyCanyon_Axis_ADCP2MHz_20160727T000005Z_20160731T235958Z-clean.csv) to the FTP server",
  "status": "running"
}
```

Property	Type	Description	Example
----------	------	-------------	---------

status	string	The current status of the data product request. Valid values are ("queued", "error" and "running")	"status": "running"
message	string	The latest status message reported by the data product process.	"message": "Transferring (BarkleyCanyon_Axis_ADCP2MHz_20160727T000005Z_20160731T235958Z-clean.csv) to the FTP server"

Cancel

The cancel method cancels the data product ran by a call to the run method.

Parameters

Parameter	Type	Description	Example
Required			
token	string	All Web Services require a token. Once logged in at http://data.oceannetworks.ca/login , your token can be retrieved or generated at http://data.oceannetworks.ca/Profile . Click on the "Web Services" tab, then click "Generate Token".	token=YOUR_TOKEN_HERE
dpRequestId	integer	A dpRequestId returned from the request method (searchHdId).	dpRequestId=1852695

Response

Success (HTTP 200)

HTTP status code 200 is returned when all of the queued and running data products in the request have been cancelled.

```
[
  {
    "dpRunId": 4521400,
    "status": "cancelled"
  }
]
```

The payload contains a list of runs, each containing the following properties:

Property	Type	Description	Example
dpRunId	integer	A Run Id that can be used to download the data product when processing is complete (searchId).	"dpRunId": 4521400
status	string	The current status of the data product request. Valid values are ("cancelled", "queued", "error", "running" and "complete").	"status": "cancelled"

Unauthorized (HTTP 401)

```
{
  "errors": [
    {
      "errorCode": 127,
      "errorMessage": "Invalid parameter value",
      "parameter": "token"
    }
  ]
}
```

errorCode	errorMessage	Description
127	Invalid parameter value	Occurs when an invalid token is used.
128	Missing parameter	Occurs when the token parameter is not included in the request

Bad Request (HTTP 400)

```
{
  "errors": [
    {
      "errorCode": 129,
      "errorMessage": "Invalid parameter name",
      "parameter": "RequestId"
    },
    {
      "errorCode": 128,
      "errorMessage": "Missing parameter",
      "parameter": "dpRequestId"
    }
  ]
}
```

errorCode	errorMessage	Description
127	Invalid parameter value	Occurs when an invalid code is used in the query. The name of the filter parameter will be included in the "parameter" property
128	Missing parameter	Occurs when multiple parameters are needed, but not all are present. Occurs when begin is used without end. The names of the required filter parameters, separated by /, are included in the parameter property,
129	Invalid parameter name	Occurs when an unsupported filter parameter is used. The name of the filter parameter will be included in the "parameter" property

Example

[https://data.oceannetworks.ca/api/dataProductDelivery?method=cancel&token=\[YOUR_TOKEN_HERE\]&dpRequestId=\[YOUR_REQUEST_ID_HERE\]](https://data.oceannetworks.ca/api/dataProductDelivery?method=cancel&token=[YOUR_TOKEN_HERE]&dpRequestId=[YOUR_REQUEST_ID_HERE])

Restart

The restart method restarts the data product cancelled by a call to the cancel method.

Parameters

Parameter	Type	Description	Example
Required			
token	string	All Web Services require a token. Once logged in at http://data.oceannetworks.ca/login , your token can be retrieved or generated at http://data.oceannetworks.ca/Profile . Click on the "Web Services" tab, then click "Generate Token".	token=YOUR_TOKEN_HERE
dpRequestId	integer	A dpRequestId returned from the request method (searchHdrId).	dpRequestId=1852695

Response

Success (HTTP 200)

HTTP status code 200 is returned when all of the cancelled data products in the request have restarted.

```
[
  {
    "dpRunId": 4521400,
    "status": "queued"
  }
]
```

The payload contains a list of runs, each containing the following properties:

Property	Type	Description	Example
dpRunId	integer	A Run Id that can be used to download the data product when processing is complete (searchId).	"dpRunId": 4521400
status	string	The current status of the data product request. Valid values are ("cancelled", "queued", "error", "running" and "complete").	"status": "queued"

Unauthorized (HTTP 401)

```
{
  "errors": [
    {
      "errorCode": 127,
      "errorMessage": "Invalid parameter value",
      "parameter": "token"
    }
  ]
}
```

errorCode	errorMessage	Description
127	Invalid parameter value	Occurs when an invalid token is used.
128	Missing parameter	Occurs when the token parameter is not included in the request

Bad Request (HTTP 400)

```
{
  "errors": [
    {
      "errorCode": 129,
      "errorMessage": "Invalid parameter name",
      "parameter": "RequestId"
    },
    {
      "errorCode": 128,
      "errorMessage": "Missing parameter",
      "parameter": "dpRequestId"
    }
  ]
}
```

errorCode	errorMessage	Description
127	Invalid parameter value	Occurs when an invalid code is used in the query. The name of the filter parameter will be included in the "parameter" property
128	Missing parameter	Occurs when multiple parameters are needed, but not all are present. Occurs when begin is used without end. The names of the required filter parameters, separated by /, are included in the parameter property,

129	Invalid parameter name	Occurs when an unsupported filter parameter is used. The name of the filter parameter will be included in the "parameter" property
-----	------------------------	--

Example

[https://data.oceannetworks.ca/api/dataProductDelivery?method=restart&token=\[YOUR_TOKEN_HERE\]&dpRequestId=\[YOUR_REQUEST_ID_HERE\]](https://data.oceannetworks.ca/api/dataProductDelivery?method=restart&token=[YOUR_TOKEN_HERE]&dpRequestId=[YOUR_REQUEST_ID_HERE])

Code Examples

Title	Creator	Modified
Ouranos Use Case	Allan Rempel	26-Feb-22
Bird Studies Canada Use Case	Allan Rempel	26-Feb-22
Internal Use Case	Ryan Ross	26-Feb-22



Please report all issues with the web services, documentation, samples and client libraries to the [Oceans 3.0 Help Centre](#)