GSFC ESDIS CMO
May 20, 2021
Released

423-ICD-015, Revision A
Earth Science Data and Information Systems (ESDIS) Project, Code 423

Cloud Notification Mechanism Interface Control Document (ICD)

National Aeronautics and
Space Administration

Goddard Space Flight Center
Greenbelt, Maryland

Cloud Notification Mechanism Interface Control Document Signature/Approval Page

Prepared by:

Signature obtained on file	***05/19/2021***
Michael Gangl PO DAAC Project System Engineer NASA JPL	Date

Reviewed by:

Signature obtained on file	***05/13/2021***
Christopher Lynnes EOSDIS System Architect GSFC Code 423	Date
Signature obtained on file	***05/13/2021***
Karen Michael EOSDIS System Manager GSFC Code 423	Date

Approved by:

Signature obtained on file	***05/19/2021***
Andrew Mitchell ESDIS Project Manager GSFC - Code 423	Date

[Electronic] Signatures available in B32 Room E148
online at: / https://ops1-cm.ems.eosdis.nasa.gov/cm2/

Preface

This document is under ESDIS Project configuration control. Once this document is approved, ESDIS approved changes are handled in accordance with Class I and Class II change control requirements described in the ESDIS Configuration Management Procedures. Changes to this document shall be verified by a document change notice (DCN) and implemented by change bars or by complete revision.

Any questions should be addressed to: esdis-esmo-cmo@lists.nasa.gov

ESDIS Configuration Management Office (CMO)
NASA/GSFC
Code 423
Greenbelt, Md. 20771

Abstract

This document identifies the Cloud Notification Mechanism (CNM) as an ESDIS standard interface mechanism. The CNM is used for the notification of ingestible data from one EOSDIS entity to another (for example a Science Data System (SDS) to a DAAC). The mechanism is push based by which messages are sent from one EOSDIS entity to another, on a commercial compute platform, using a push notification service (in the case of Amazon Web Services, Simple Notification Service (SNS), Simple Queue Service (SQS), or Kinesis). The messages detail a product along with the files it is comprised of. The CNM allows for an optional response message from the receiving entity to the provider notifying of success or error of the previously received message.

Keywords: SIPS, DAAC, CNM, SNS, AWS

Change History Log

Revision	Effective Date	Description of Changes (Reference the CCR & CCB Approval Date)
Original	07/18/2018	CCR 423-ESDIS-179; CCB Approved 06/25/2018 Pages: Cover through 12
Revison A	05/19/2021	CCR 423-ESDIS-322; CCB Approved 04/29/2021 Pages: Cover through page 15

Table of Contents

List of Figures

List of Tables

1 INTRODUCTION

1.1 Purpose

The Polling with Delivery Record (PDR) mechanism has served Earth Observing System Data and Information System (EOSDIS) well for quite a while, providers and data created in the cloud have an array of tools and services available of which we should take advantage. As ingest and archive systems migrate to commercial cloud platforms, they become more scalable and performant, an updated transfer scheme is needed to enable that performance and scalability as well as replace older or unsupported transfer protocols, such as Secure File Transfer Protocol (SFTP) and Secure Copy Protocol (SCP). This is the purpose of the Cloud Notification Mechanism (CNM).

1.2 Scope

This document was developed to be used as a standard protocol for exchange of data between EOSDIS elements in the cloud. The CNM is to be used inside the cloud for very high bandwidth handshaking using Amazon S3 (Simple Storage Service) buckets. While designed for Amazon Web Services (AWS) cloud-to-cloud based transfers to be used by upcoming missions and data streams, any commercial cloud platform could take advantage of the CNM schema.

1.3 Related Documentation

The latest versions of all documents below should be used. The latest ESDIS Project documents can be obtained from URL: https://ops1-cm.ems.eosdis.nasa.gov. ESDIS documents have a document number starting with either 423 or 505. Other documents are available for reference in the ESDIS project library website at: https://doclib.eosdis.nasa.gov/ unless indicated otherwise.

1.3.1 Applicable Documents

The following documents are referenced within, are directly applicable, or contain policies or other directive matters that are binding upon the content of this document.

423-41-57	ICD between the ECS and the SIPS, Volume 0, Interface Mechanisms

2 OVERVIEW

2.1 Key Characteristic of the Cloud Notification Mechanism

The defining characteristics of the CNM are as follows:

- Pure push mechanism, no polling by Distributed Active Archive Centers (DAACs) or Science Investigator-led Processing System (SIPS)- reduces latency in availability notifications of products, and also necessary to avoid listing the contents of an S3 bucket which can be expensive and slow depending on bucket size.
- Single product notification, (e.g. a single granule per message).
- Success response messages are optional
- JSON formatted- JSON is ubiquitous amongst most languages
- Stripped down required fields and overloaded data types

2.2 Summary of Data Flows

2.2.1 Prerequisites

Both the provider and consumer of the CNM messages must agree on the push notification service (e.g. a kinesis stream name, SNS topic, or SQS Queue) and have the authorization to retrieve from or publish messages to the service. This is true for both the submissions and response notifications.

2.2.2 Summary of Data Flow Steps

The following Figure 2-1 CNM data flow diagram is used to portray all the steps involved or the "handshake" for the exchange of data in the cloud.

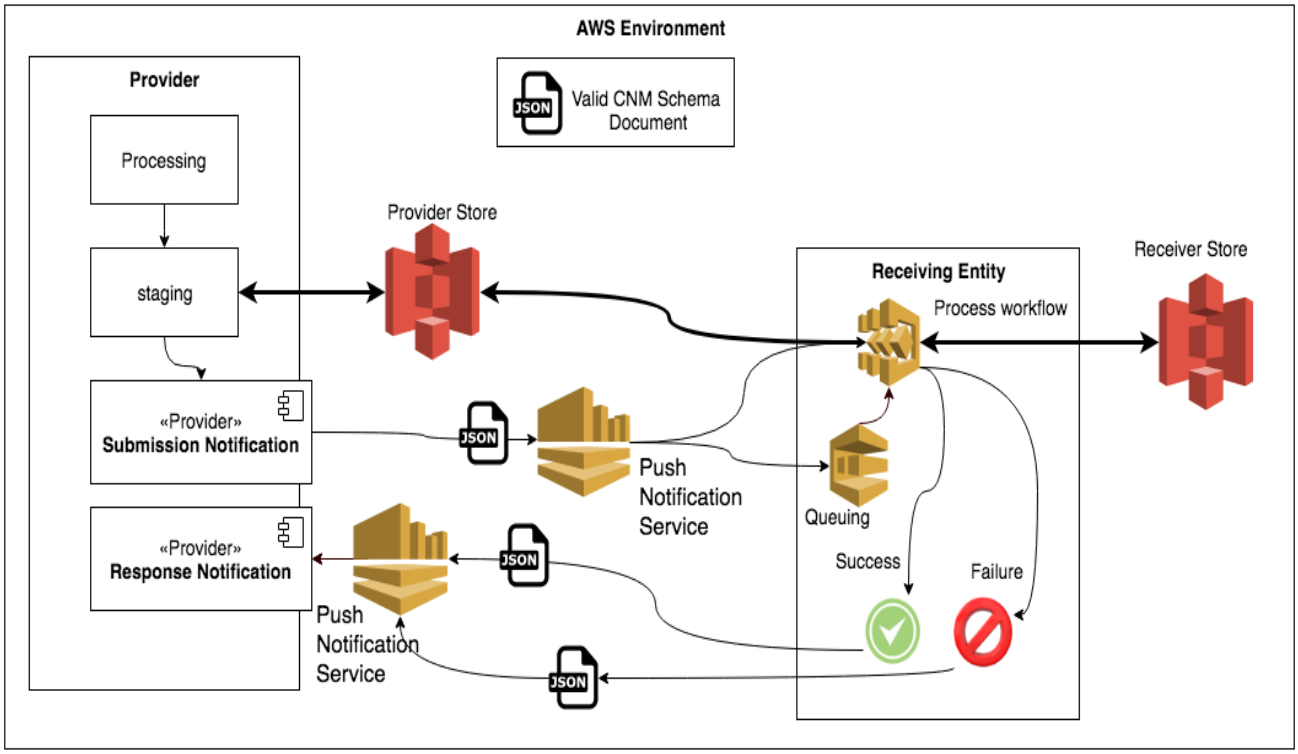

Figure 2-1. CNM Data Flows.

1. A provider processes data files and stages them in one or more online storage locations (e.g. an AWS bucket). Files for a single product can reside in multiple buckets, which may be advantageous for storage lifecycling, versioning, or other reasons.

2. The provider generates a CNM message *for each product* comprised of one or more files to be ingested. Each message is assigned a unique message ID by the providing entity.

3. The CNM message is sent to a push notification service that has been previously agreed to by both entities.

4. The CNM message is received by the receiving entity from the notification service and either immediately processed or queued for eventual processing.

5. The receiving entity executes a process workflow using the message to determine the location of product files, and retrieve these files from the provider's data store, writing them to the receiving entity's storage location.

6. Workflows continue processing the product as per nominal ingestion.

7. Upon successful or failed ingestion, a CNM response messages is sent to a previously agreed upon push notification service detailing the message ID and the status message. In the event of a processing error, the CNM Response message will contain the appropriate error code and message.

2.3 Message Specifications

The below specifications define version 1.5.1 of the CNM schema. There are two types of messages valid in the CNM. The first message is the notification of new data. This message is called the CNM submission, or CNM-S. The second is a CNM response (CNM-R) message, alerting the sending entity to the status of the CNM-S. Both CNM-S and CNM-R are JSON formatted messages.

2.3.1 Notification of New Data

Table 2-1. Notification Fields

Property	Required	Definition	Notes
version	yes	The version of the CNM schema being used.	All 1.X versions should be backwards compatible. A major version release (e.g. 2.X.Y) cannot ensure backwards compatibility.
submissionTime	yes	The ISO8601 datetime the message was created (and presumably sent) to the publication mechanism. GMT Required.	
identifier	yes	a provider generated unique identifier for this message.	This shall not be reused
collection	yes	The collection to which the product belongs.	This may be used if a generic publication mechanism is used for multiple providers. Can include the collection version as well e.g. MODIS/1
provider	no	The provider/SIPS identifier which created this message.	
product	yes	A product object, defined below. Only a single product, composed of multiple files, can be defined in each message.	Could be a granule or some other product
trace	no	Information on the message or who is sending it.	

Table 2-2. Product Object Fields

Property	Required	Definition	Notes
name	yes	The name of the product to be processed.	For a granule, typically the GranuleUR
dataVersion	yes	version number/string of the product.	
files	yes	an array of file objects that make up the product.	See the file::* entries for properties of a file object. One of files or filegroups are required.
filegroups	yes	Array of filegroups that make up this group	See the filegroup::* entries for properties of a file object. One of files or filegroups are required.
filegroups::id	yes	String id of the filegroup by which all files are associated	
filegroups::files	yes	an array of file objects that make up the product.	
file::type	yes	The type of file. science files (netcdf, HDF, binary) should use the 'data' type. More can be added if need and consensus demand.	Data types are defined in the File Types' table below.
file::subtype	no	An optional, specific implementation of the file::type. e.g. NetCDF for a file of type 'data'	
file::uri	yes	The URI of the provided file (e.g. s3://...)	
file::name	no	The name of the file.	The file::uri provided within the file object may utilize a hash or other randomized naming convention. This field allows for a more human readable file name to be used when the receiving entity writes the file to its data store.
file::checksumType	yes	Type of the checksum (e.g. md5).	
file::checksum	yes	Checksum of the file.	
file::size	yes	Size, in bytes, of the file.	

Table 2-3. File Types

Data Type	Example Files	Notes
data	NetCDF files, HDF file, binary science files	
browse	png, jpeg, GIBS browse images	
metadata	checksums, ODL or XML, production history	
ancillary linkage	Quality Assessment files, ph, ncml, fmt file describe	

2.3.2 Optional Response Message

The response for successful granules is OPTIONAL per the ICD between the SIPS and DAAC or the processing Element and the Archive Element. It is recommended that the response message be used in cases of both success and failures, even though success responses are optional.

2.3.3 Best Practices

Ownership of publication mechanism is up to the entities exchanging messages. It may be beneficial for the provider to own both the submission mechanism as well as the response mechanism, and grant access via account or role.

A single publication mechanism should be able to support multiple collection types. That is, a SIPS or Science Data System (SDS) may send messages for multiple collections to a single end-point. See below image.

The response message should be used in cases of both success and failures, even though success responses are optional.

Required fields are minimal in the CNM specification. The expectation is that entities passing CNM messages will define more fields as required for their use case.

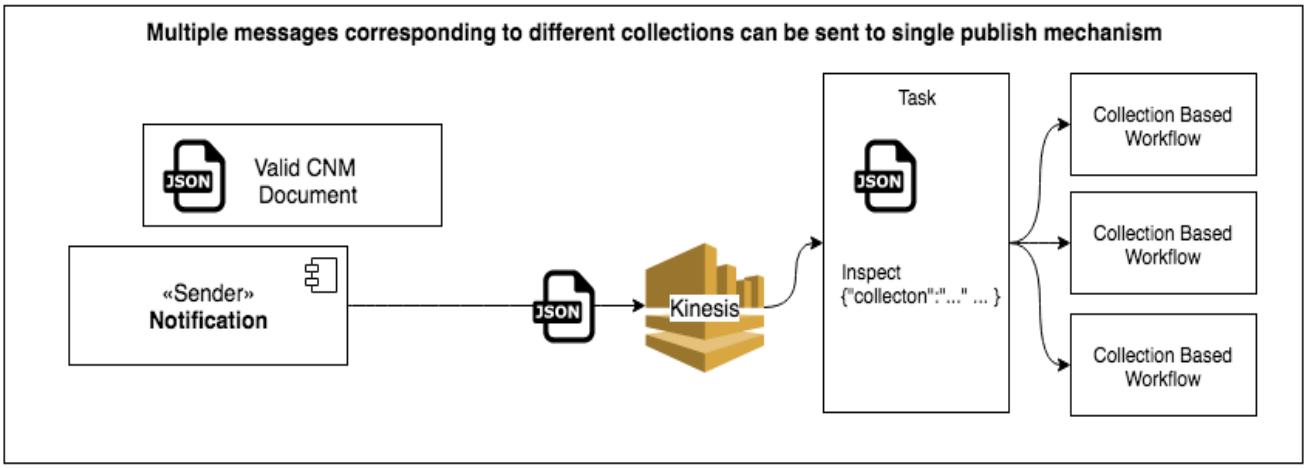

Figure 2-2. Flow of Multiple Messages for Different Collections.

Table 2-4. Response Message Fields (1 of 2)

Property	Required	Definition	Notes
version	yes	The version of the CNM schema being used.	This document controls and describes all schema versions.
submissionTime	yes	The ISO8601 datetime the message was originally sent by the provider. GMT Required.	In a response, this field should be copied from the incoming message.
receivedTime	yes	The ISO8601 datetime the message was received by the receiving entity. GMT Required.	This will nominally be very close to the submission time, but there are some cases where it may be significantly later. If a queue is backing the publication mechanism, the received time should be set to when the DAAC started processing once removed from the queue.
processCompleteTime	yes	The ISO8601 datetime processing completed by the receiving entity. GMT Required.	
identifier	yes	The generated unique identifier for the CNM-S message.	Must be kept from the original CNM-S.
collection	yes	The collection for which the product belongs.	This may be used if a generic publication mechanism for multiple providers.
trace	no	Info on who is sending the message.	Useful when expecting multiple success messages or when some information wants to be made available to the receiver of the CNM-R.
response	yes	A response object, defined below.	

Table 2-5. Response Message Fields (2 of 2)			
Property	**Required**	**Definition**	**Notes**
product	no	See ‘product’ fields defined in table 2.2 above	A success or failure response can have an optional ‘product’ element. For the general case of CNM, a response is sent as a success if the receiver was able to successfully ingest one or more files specified in the original message. A failure is sent if some error occurred while trying to ingest one or more files specified in the original message. The response is sent, generally, to the data provider who sent the original CNM message, thought it could be sent to others. The original consumer may rely on the archive location for future processing, and as such this ‘product’ field can specify the final archive location as well as the CMR concept-id of the product. Specifying this information can help data providers avoid the lookup of data/location in CMR. If this is provided, it is highly recommended to also implement the “ ingestionMetadata” fields of the ‘response’ object to provide lookup information of the granule in the event that data are moved or endpoints are changed- that is, the location of the data in a response object might not be a valid location in the future.

Table 2-6. Response Object Fields

Property	Required	Definition	Notes
status	yes	"SUCCESS" or "FAILURE"	
errorCode	no	"VALIDATION_ERROR", "TRANSFER_ERROR", or "PROCESSING_ERROR"	Not valid, unless this respone is for an error
errorMessage	no	A more descriptive message for failure than the above code. This could be a stack trace or a human readable message: file size mismatch, or "file not found on host system".	Not required, unless this response is for an error
ingestionMetadata	no	Object defining ingestion metadata, like CMR concept IDS, URLS, etc.	See ingestionMetadata::* entries for more information
ingestionMetadata::catalogId	no	Identifier for a catalog entry	For example, CMR granule concept-id
ingestionMetadata::catalogURL	no	URL of a catalog entry	

Table 2-7. Response Error Codes

ErrorCode	Definition
VALIDATION_ERROR	Validation error: This error is reported in the following situations: 1) CNM-S validation failure (e.g. schema non-compliance) 2) Product file validation failure (e.g. invalid checksum) 3) Product business logic failure (e.g. missing or incorrect metadata file)
TRANSFER_ERROR	Transfer error: This error is reported in the following situations: 1) Receiver cannot access one or more Product:file 2) Receiver experienced issues transferring one or more Product:file
PROCESSING_ERROR	Processing error: This error is reported in the following situations: 1) Internal processing error by the receiver

3 NOTIFICATION MESSAGE EXAMPLES

For the most up-to-date and exhaustive examples, see https://github.com/podaac/cloud-notification-message-schema/tree/master/samples

3.1 Notification Messages


```
"version": "1.0"
"provider": "PODAAC_SWOT",
"collection": "SWOT_Prod_l2:1",
"submissionTime":"2017-09-30T03:42:29.791198",
"identifier": "1234-abcd-efg0-9876",
"product":
{
  "name": "sampleGranuleName001",
  "dataVersion": "001",
  "files": [
    {
      "type": "data",
      "uri": "s3://myIngestBucket/prod_20170926T11:30:36/production_file.nc",
      "name":"production_file.nc",
      "checksumType": "md5",
      "checksum": "b1d1a645a3148553936cf13bcafd75bb",
      "size": 123456
    },
    {
      "type": "browse",
      "uri": "s3://myIngestBucket/prod_20170926T11:30:36/production_file.png",
      "name":"production_file.png",
      "checksumType": "md5",
      "checksum": " b1d1a645a3148553936cf13bcafd75bc",
      "size": 12345
      }
  ]
 }
}
```


Figure 3-1. Example CNM Notification from SIPS to DAAC

```
{
 "version": "1.0"
 "provider": "PODAAC_SWOT",
 "collection": "SWOT_Prod_l2:1",
 "submissionTime":"2017-09-30T03:42:29.791198",
 "identifier": "1234-abcd-efg0-9876",
 "product":
 {
  "name": "sampleGranuleName001",
  "dataVersion": "001",
  "files": [
   {
    "type": "data",
    "uri": "s3://myIngestBucket/prod_20170926T11:30:36/production_file.nc",
    "name":"production_file.nc",
    "checksumType": "md5",
    "checksum": " b1d1a645a3148553936cf13bcafd75bb",
    "size": 123456
   },
   {
    "type": "browse",
    "uri": "s3://ADifferentIngestBucket/prod_20170926T11:30:36/production_file.png",
    "name":"production_file.png",
    "checksumType": "md5",
    "checksum": " b1d1a645a3148553936cf13bcafd75bc",
    "size": 12345
   }
  ]
 }
}
```


Figure 3-2. Example CNM Notification from SIPS to DAAC, files in different buckets

```
{
  "version":"1.1",
  "provider": "PODAAC_SWOT",
  "collection": "SWOT_Prod_l2:1",
  "submissionTime":"2017-09-30T03:42:29.791198Z",
  "identifier": "1234-abcd-efg0-9876",
  "product":
    {
      "name": "sampleGranuleName001",
      "dataVersion": "001",
      "filegroups" : [{
        "id" : "filgroup1",
        "files": [
          {
            "type": "data",
            "uri": "s3://sampleIngestBucket/prod_20170926T11:30:36/production_file.nc",
            "name":"production_file.nc",
            "checksumType": "md5",
            "checksum": "4241jafkjaj14jasjf",
            "size": 123456
          },
          {
            "type": "browse",
            "uri": "s3://sampleIngestBucket/prod_20170926T11:30:36/production_file.png",
            "name":"production_file.png",
            "checksumType": "md5",
            "checksum": "addjd872342bfbf",
            "size": 12345
          }
        ]
      },
      {
        "id" : "filgroup2",
        "files": [
          {
            "type": "data",
            "uri": "s3://sampleIngestBucket/prod_20170926T11:30:36/production_file.nc",
            "name":"production_file.nc",
            "checksumType": "md5",
            "checksum": "4241jafkjaj14jasjf",
            "size": 123456
          },
          {
            "type": "browse",
            "uri": "s3://sampleIngestBucket/prod_20170926T11:30:36/production_file.png",
            "name":"production_file.png",
            "checksumType": "md5",
            "checksum": "addjd872342bfbf",
            "size": 12345
          }
        ]
      }]
    }
}
```


Figure 3-3. Example CNM message utilizing filegroups.

3.2 Response Messages


```
{
 "version": "1.0"
 "provider": "PODAAC_SWOT",
 "collection": "SWOT_Prod_l2:1",
 "processCompleteTime":"2017-09-30T03:45:29.791198",
 "receivedTime":"2017-09-30T03:42:31.634552",
 "submissionTime":"2017-09-30T03:42:29.791198",
 "identifier": "1234-abcd-efg0-9876",
 "trace":"metadata extraction workflow"
 "response": {
  "status":"SUCCESS"
 }
}
```


Figure 3-4. Example CNM Success Response


```
{
 "version": "1.0"
  "provider": "PODAAC_SWOT",
  "collection": "SWOT_Prod_l2:1",
  "processCompleteTime":"2017-09-30T03:45:29.791198",
  "submissionTime":"2017-09-30T03:42:29.791198",
  "receivedTime":"2017-09-30T03:42:31.634552",
  "identifier": "1234-abcd-efg0-9876",
  "trace":"metadata extraction workflow"
  "response": {
    "status":"FAILURE",
    "errorCode": "VALIDATION_ERROR",
    "errorMessage": "File [cumulus-dev-a4d38f59-5e57-590c-a2be-
58640db02d91/prod_20170926T11:30:36/production_file.nc] did not match given checksum
value."
  }
```


Figure 3-5. Example CNM Error Response

```
{
  "version": "1.5.1",
  "provider": "PODAAC",
  "submissionTime": "2020-11-06T17:17:29.339531Z",
  "collection": "MODIS_A-JPL-L2P-v2019.0",
  "identifier": "f2d5aca2-2053-11eb-8e6f-acde48001122",
  "trace": "NCMODIS_A-JPL-L2P-v2019.01",
  "product": {
    "dataVersion": "2019.0",
    "files": [
      {
        "type": "data",
        "name": "20200101000000-JPL-L2P_GHRSST-SSTskin-MODIS_A-D-v02.0-fv01.0.nc",
        "uri": "https://es3uvapgyg.execute-api.us-west-2.amazonaws.com:9000/DEV/podaac-sndbx-
cumulus-protected/MODIS_A-JPL-L2P-v2019.0/20200101000000-JPL-L2P_GHRSST-SSTskin-MODIS_A-D-
v02.0-fv01.0.nc",
        "checksumType": "md5",
        "checksum": "aa5204f125ae83847b3b80fa2e571b00",
        "size": 18232098
      },
      {
        "type": "metadata",
        "name": "20200101000000-JPL-L2P_GHRSST-SSTskin-MODIS_A-D-v02.0-fv01.0.nc.md5",
        "uri": "https://es3uvapgyg.execute-api.us-west-2.amazonaws.com:9000/DEV/podaac-sndbx-
cumulus-public/MODIS_A-JPL-L2P-v2019.0/20200101000000-JPL-L2P_GHRSST-SSTskin-MODIS_A-D-v02.0-
fv01.0.nc.md5",
        "size": 98
      },
      {
        "type": "metadata",
        "name": "20200101000000-JPL-L2P_GHRSST-SSTskin-MODIS_A-D-v02.0-fv01.0.cmr.json",
        "uri": "https://es3uvapgyg.execute-api.us-west-2.amazonaws.com:9000/DEV/podaac-sndbx-
cumulus-public/MODIS_A-JPL-L2P-v2019.0/20200101000000-JPL-L2P_GHRSST-SSTskin-MODIS_A-D-v02.0-
fv01.0.cmr.json",
        "size": 1633
      }
    ],
    "name": "20200101000000-JPL-L2P_GHRSST-SSTskin-MODIS_A-D-v02.0-fv01.0"
  },
  "receivedTime": "2020-11-06T17:18:49.506Z",
  "response": {
    "status": "SUCCESS",
    "ingestionMetadata": {
      "catalogId": "G1238611022-POCUMULUS",
      "catalogUrl": "https://cmr.uat.earthdata.nasa.gov/search/granules.json?
concept_id=G1238611022-POCUMULUS"
    }
  },
  "processCompleteTime": "2020-11-06T17:21:13.523Z"
}
```


Figure 3-6. CNM Response with 'ingestionMetadata' fields

Appendix A Abbreviations and Acronyms

AWS	Amazon Web Services
CCB	Configuration Change Board
CCR	Configuration Change Request
CMO	Configuration Management Office
CNM	Cloud Notification Mechanism
CNM-R	Cloud Notification Mechanism – Response
CNM-S	Cloud Notification Mechanism - Submission
DAAC	Distributed Active Archive Center
DCN	Document Change Notice
ECS	EOSDIS Core System
EOSDIS	Earth Observing System Data and Information System
ESDIS	Earth Science Data Information System
GIBS	Global Image Browse Service
GSFC	Goddard Space Flight Center
HDF	Hierarchical Data Format
ICD	Interface Control Document
JPL	Jet Propulsion Laboratory
JSON	JavaScript Object Notation
MD5	Message Digest 5
MODIS	Moderate Resolution Imaging Spectroradiometer (EOS)
NASA	National Aeronautics and Space Administration
NeTCDF	Network Common Data Form
ODL	Object Definition Language
PDR	Polling with Delivery Record
PNG	Portable Network Graphics
PO DAAC	Physical Oceanography Data Active Archive and Catalog
S3	Simple Storage Service
SCP	Secure Copy Protocol
SDS	Science Data System
SFTP	Secure File Transfer Protocol
SIPS	Science Investigator-led Processing System
SNS	Simple Notification Service
SQS	Simple Queue Service
TBD	To Be Determined
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
XML	Extensible Markup Language