



LANDFIRE's Request Validation Web Service User's Guide

LANDFIRE's large mosaics can be clipped out in portions using user-specific geographic limits. This functionality includes a set of web services for the developer that can be incorporated into custom applications.

- **Request Validation Web Service** verifies and validates the dataset information and a user-defined area of interest and provides a fully parameterized URL(s) that can be used by the Download Service.
- **Download Web Service** initiates a request for products, queries the system to obtain a job status, and returns the requested products to the user.

This document discusses how to use the Request Validation web service to generate valid product requests.

Request Validation Web Service

The Request Validation Service is used to generate download requests for LANDFIRE mosaics by performing the following tasks:

- Verify that the total area of interest is of a reasonable/downloadable size. For instance, if the bounding coordinates for the entire contiguous United States were submitted to the Download Service, an error message would be received because that area would exceed the download service limit of 1.5 gigabytes (estimated). This is to prevent the service from creating an unmanageable number of download and thumbnail URLs at one time. Although, there is no limit as to the number of times a user can call the Request Validation web service, there is a limit to the size of each request. This is to keep the service from wasting time building URLs that may never actually be used.
- Break up the area of interest into more manageable pieces for downloading and usability purposes. A 1.5 gigabyte file would take quite a while to download and may be too large to load into a client application, so the area of interest is broken up into equal size pieces between 15 and 250 megabytes. Each resulting piece must be requested and downloaded separately when retrieving products.
- Validates that the product key, output format, metadata format, and compression format are valid combinations for a dataset. All requests must be submitted to the Download Service for the actual download of the products.

The URL to the Request Validation Service WSDL page is:

<https://landfire.cr.usgs.gov/requestValidationService/wsd/RequestValidationService.wsdl>

Notes on Request Limits

The maximum size of an individual download is currently set at 250 MB. Smaller pieces can be requested by using the "CHUNK_SIZE" parameter in the processAOI and processAOI2 methods. Requests larger than the desired chunk size will be recursively subdivided until each piece is less than the chunk size.



There is also a limit to the total size (in MB) that is allowed by the Request Validation service. If the estimated size for all products combined is greater than 1.5 GB, then an error message will be returned. This limitation is to prevent an extremely large and unmanageable number of links being returned by the service at one time.

LAYER_IDS Tag Description

Each of the four Request Validation Service methods require an input parameter called "LAYER_IDS". This section describes how to construct this string.

For mosaics download requests, the data is clipped in real time and the download bundle is constructed per the user's desired options. Therefore, the LAYER_IDS string must be constructed based on the desired output format, metadata format, and file compression method. Once the format the products should be returned as is selected, those codes need to be appended onto the end of the productkey.

For example, if we want to download "us_110 Existing Vegetation Type" in ArcGRID_with_attris (08) format and the metadata in xml (X) format, compressed in a zip (Z) file, our string would look like F4M08XZ. The order is: productkey (always 3 char), output format (always 2 char), metadata format (always 1 char), and file compression format (always 1 char). Therefore, we would use F4M08XZ for the LAYER_IDS string: <LAYER_IDS>F4M08XZ</LAYER_IDS>

Error Messages

This example shows the xml tags used when returning an error message from this web service:

```
<ERROR>
  <FIELD_NAME>Udaoi</FIELD_NAME>
  <ERROR_MSG>User defined area of interest greater than max allowed:
26366.855377122745 vs 1500.0</ERROR_MSG>
</ERROR>
```

This example shows the JSON string that will be used when returning any error from this web service:

```
{"REQUEST_SERVICE_RESPONSE":{"ERROR":{"FIELD_NAME":"Udaoi","ERROR_MSG":"User
defined area of interest greater than max allowed: 276847.13340001047 vs
1500.0"},"STATUS":false}}
```

processAOI

This is the first method that was originally provided in the service. The processAOI method only takes one parameter - an xml formatted string like this:

```
<REQUEST_SERVICE_INPUT>
  <AOI_GEOMETRY>
    <EXTENT>
      <TOP></TOP>
      <BOTTOM></BOTTOM>
      <LEFT></LEFT>
      <RIGHT></RIGHT>
    </EXTENT>
    <SPATIALREFERENCE_WKID/>
  </AOI_GEOMETRY>
```



```

<LAYER_INFORMATION>
  <LAYER_IDS> </LAYER_IDS>
</LAYER_INFORMATION>
<CHUNK_SIZE></CHUNK_SIZE>
<ORIGINATOR/>
<JSON></JSON>
</REQUEST_SERVICE_INPUT>

```

This method returns a string containing two tags for each download piece – a `DOWNLOAD_URL` and a `THUMBNAIL_URL`. The `DOWNLOAD_URL` is used to call the Download Service to physically download products. The `THUMBNAIL_URL` can be used at any time to view a down-sampled image of the requested piece. Please note that this method will continue to be supported but has been superseded by the newer `processAOI2` method.

processAOI2

This method is an updated version of the original `processAOI` method. The input remains the same - an xml formatted string like this:

```

<REQUEST_SERVICE_INPUT>
  <AOI_GEOMETRY>
    <EXTENT>
      <TOP></TOP>
      <BOTTOM></BOTTOM>
      <LEFT></LEFT>
      <RIGHT></RIGHT>
    </EXTENT>
    <SPATIALREFERENCE_WKID/>
  </AOI_GEOMETRY>
  <LAYER_INFORMATION>
    <LAYER_IDS> </LAYER_IDS>
  </LAYER_INFORMATION>
  <CHUNK_SIZE></CHUNK_SIZE>
  <ORIGINATOR/>
  <JSON></JSON>
</REQUEST_SERVICE_INPUT>

```

This method returns a string containing an additional tag (`METADATA_URL`) for each download piece to go along with the `DOWNLOAD_URL` and `THUMBNAIL_URL` tags.

Sample Java code for calling the processAOI2() Method

```

String validationServiceEndpoint =

"https://landfire.cr.usgs.gov/requestValidationService/services/RequestValidationS
ervice";
String validationServiceMethod = "processAOI2";
String xmlRequestString = "<REQUEST_SERVICE_INPUT>" +
  "<AOI_GEOMETRY>" +
    "<EXTENT>" +
      "<TOP> 40.840</TOP>" +
      "<BOTTOM>40.815</BOTTOM>" +
      "<LEFT>-96.715</LEFT>" +
      "<RIGHT>-96.689</RIGHT>" +

```



```

        "</EXTENT>" +
        "<SPATIALREFERENCE_WKID/>" +
    "</AOI_GEOMETRY>" +
    "<LAYER_INFORMATION>" +
        "<LAYER_IDS>F4M08HZ</LAYER_IDS>" +
    "</LAYER_INFORMATION>" +
    "<CHUNK_SIZE>250</CHUNK_SIZE>" +
    "<ORIGINATOR/>" +
    "<JSON>>false</JSON>" +
"</REQUEST_SERVICE_INPUT>";

// Make request
try {
    Service validationService = new org.apache.axis.client.Service();
    Call validationCall = (Call) validationService.createCall();
    validationCall.setTargetEndpointAddress(new
java.net.URL(validationServiceEndpoint));
    validationCall.setOperationName( new QName("https://edc.usgs.gov",
        validationServiceMethod) );
    String serviceResponse = (String) validationCall.invoke( new Object[] {new
String(xmlRequestString) } );
} catch (Exception e) {
    System.out.println("Error calling ValidationService: " + e.toString());
}

// Process serviceResponse here

```

Sample Python code for calling the processAOI2() Method

```

wsdlUrlValidationService =
'https://landfire.cr.usgs.gov/requestValidationService/wsdl/RequestValidationService.wsdl'

try:
    #Query the Validation Service
    server = ServiceProxy(wsdl=wsdlUrlValidationService)

    xmlRequestString =
"<REQUEST_SERVICE_INPUT><AOI_GEOMETRY><EXTENT><TOP>40.840</TOP><BOTTOM>40.815</BOTTOM><LEFT>-96.715</LEFT><RIGHT>-96.689</RIGHT></EXTENT><SPATIALREFERENCE_WKID/></AOI_GEOMETRY><LAYER_INFORMATION><LAYER_IDS>F4M08HZ</LAYER_IDS></LAYER_INFORMATION><CHUNK_SIZE>250</CHUNK_SIZE><JSON></JSON></REQUEST_SERVICE_INPUT>"
    processAOI2ResponseDict =
server.processAOI2(requestInfoXml=xmlRequestString)
    processAOI2Response = processAOI2ResponseDict['serviceResponse']

except:
    processAOI2Response = ""
    #raise

# Process processAOI2Response here

```



REST Information

REST commonly refers to a stateless client server architecture whereby web services, such as the USGS Request Validation Service, can be accessed via a simple URI. So instead of obtaining information from the Request Validation Service using the two methods listed above (processAOI, processAOI2), the same information can be retrieved using a simple http request.

A. processAOI

Sample REST request for LANDFIRE seamless data:

```
https://landfire.cr.usgs.gov/requestValidationServiceClient/sampleRequestValidationServiceProxy/processAOI.jsp?TOP=39.825&BOTTOM=39.824&LEFT=-105.091&RIGHT=-105.090&LAYER_IDS=F4M08HZ&JSON=true
```

Response:

```
{"REQUEST_SERVICE_RESPONSE":{"PIECE":[{"THUMBNAIL_URL":"https://landfire.cr.usgs.gov/servlet/com.esri.wms.Esrimap?servicename=USGS_EDC_LandFire&Version=1.1.1&SERVICE=WMS&request=map&layers=LANDFIRE.US_110EVT&bbox=-105.091,39.824,-105.09,39.825&reaspect=false&width=39&height=40&format=jpeg&SRS=EPSG:4326&styles=","DOWNLOAD_URL":"https://landfire.cr.usgs.gov/axis2/services/DownloadService/initiateDownload?siz=1&key=F4M&ras=1&pfm=ArcGRID_with_attribs&imsurl=-1&ms=-1&att=-1&lay=-1&fid=-1&dlpre=lf&lft=-105.091&rgt=-105.09&top=39.825&bot=39.824&wmd=1&mur=https://landfire.cr.usgs.gov/distmeta/servlet/gov.usgs.edc.MetaBuilder&mcd=F4M&mdf=HTML&arc=ZIP&sde=LANDFIRE.US_110EVT&msd=LANDFIRE.US_110_SPATPLYGN_MASTER&zun=METERS&prj=102039&rsp=0&bnd=&bndnm=&csx=30.0&csy=30.0&ics=&ORIG=RVS"}],"STATUS":true}}
```

B. processAOI2

Sample REST request for LANDFIRE seamless data:

```
https://landfire.cr.usgs.gov/requestValidationServiceClient/sampleRequestValidationServiceProxy/processAOI2.jsp?TOP=29.426617875735545&BOTTOM=29.420069397917526&LEFT=-98.49394726686275&RIGHT=-98.48708695676768&LAYER_IDS=F4M08HZ&CHUNK_SIZE=250&JSON=true
```

Response:

```
{"REQUEST_SERVICE_RESPONSE":{"PIECE":[{"THUMBNAIL_URL":"https://landfire.cr.usgs.gov/servlet/com.esri.wms.Esrimap?servicename=USGS_EDC_LandFire&Version=1.1.1&SERVICE=WMS&request=map&layers=LANDFIRE.US_110EVT&bbox=-98.49394726686275,29.420069397917526,-98.48708695676768,29.426617875735545&reaspect=false&width=40&height=38&format=jpeg&SRS=EPSG:4326&styles=","METADATA_URL":"https://landfire.cr.usgs.gov/distmeta/servlet/gov.usgs.edc.MetaBuilder?TYPE=HTML&DATASET=F4M&YMAX=29.426617875735545&YMIN=29.420069397917526&XMIN=-98.48708695676768&XMAX=-98.49394726686275&coordsys=0","DOWNLOAD_URL":"https://landfire.cr.usgs.gov/axis2/services/DownloadService/initiateDownload?siz=1&key=F4M&ras=1&pfm=ArcGRID_with_attribs&imsurl=-1&ms=-1&att=-1&lay=-1&fid=-1&dlpre=lf&lft=-98.49394726686275&rgt=-98.48708695676768&top=29.426617875735545&bot=29.420069397917526&wmd=1&mur=https://landfire.cr.usgs.gov/distmeta/servlet/gov.usgs.edc.MetaBuilder&mcd=F4M&mdf=HTML&arc=ZIP&sde=LANDFIRE.US_110EVT&msd=LANDFIRE.US_110_SPATPLYGN_MASTER&zun=METERS&prj=102039&rsp=0&bnd=&bndnm=&csx=30.0&csy=30.0&ics=&ORIG=RVS"}],"STATUS":true}}
```



Acronyms and Abbreviations

Acronym	Definition
WMS	Web Map Service
URI	Uniform Resource Identifier
REST	Representational State Transfer
WSDL	Web Service Description Language