RDP Generic CFS File Workflow

Last Update: June 2025

Importing libararies

In [1]:
```
import os
import sys
import requests
import json
```

Set RDP credentials and Initial Parameters

In [ ]:
```
username = 'your RDP Machine-ID'
password = 'your RDP Password'
clientId = 'your RDP APP Key'

RDP_HOST= 'https://api.refinitiv.com'
acccess_token = None
refresh_token = None
expires_in = 0
```

RDP APIs Application Workflow

Step 1: Authentication with RDP APIs

RDP APIs entitlement check is based on OAuth 2.0 specification. The first step of an application workflow is to get a token from RDP Auth Service, which will allow access to the protected resource, i.e. data REST API.

The API requires the following access credential information:

- Username: The username.
- Password: Password associated with the username.
- Client ID: This is also known as `AppKey`, and it is generated using an App key Generator. This unique identifier is defined for the user or application and is deemed

confidential (not shared between users). The client_id parameter can be passed in the request body or as an "Authorization" request header that is encoded as base64.

- Grant Type `password` : This is for initial authentication request. An application does not have any token, so it requests new tokens using username/password combination.

The HTTP request for the RDP APIs Authentication service is as follows:

```
POST /auth/oauth2/v1/token HTTP/1.1
Accept: */*
Content-Type: application/x-www-form-urlencoded
Host: api.refinitiv.com:443
Content-Length: XXX

username=RDP_USERNAME
&password=RDP_PASSWORD
&client_id=RDP_APP_KEY
&grant_type=password
&takeExclusiveSignOnControl=true
&scope=trapi
```

Once the authentication success, the function gets the RDP Auth service response message and keeps the following RDP token information in the variables.

- **access_token**: The token used to invoke REST data API calls as described above. The application must keep this credential for further RDP APIs requests.
- **refresh_token**: Refresh token to be used for obtaining an updated access token before expiration. The application must keep this credential for access token renewal.
- **expires_in**: Access token validity time in seconds.

Next, after the application received the Access Token (and authorization token) from RDP Auth Service, all subsequent REST API calls will use this token to get the data. Please find more detail regarding RDP APIs workflow in the following resources:

- RDP APIs: Introduction to the Request-Response API page.
- RDP APIs: Authorization - All about tokens page.

In [ ]:
```
#step 1 - get RDP Access Token from RDP

# Send HTTP Request
auth_url = f'{RDP_HOST}/auth/oauth2/v1/token'
payload = f'grant_type=password&username={username}&client_id={clientId}&password={
try:
    auth_response = requests.post(auth_url,
                                  headers = {'Content-Type':'application/x-www-form-urle
                                  data = payload,
                                  auth = (clientId, '')
                )
except requests.exceptions.RequestException as exp:
    print(f'Caught exception: {exp}')
```

```
if auth_response.status_code == 200:  # HTTP Status 'OK'
    print('Authentication success')
    access_token = auth_response.json()['access_token']
    refresh_token = auth_response.json()['refresh_token']
    expires_in = int(auth_response.json()['expires_in'])

if auth_response.status_code != 200:
    print(f'RDP authentication failure: {auth_response.status_code} {auth_response.
    print(f'Text: {auth_response.text}')
```

```
Authentication success
```

Requesting Data from RDP APIs

That brings us to requesting the RDP APIs data. All subsequent REST API calls use the Access Token via the *Authorization* HTTP request message header as shown below to get the data.

- Header:
  - Authorization = `Bearer <RDP Access Token>`

Please notice *the space* between the `Bearer` and `RDP Access Token` values.

The application then creates a request message in a JSON message format or URL query parameter based on the interested service and sends it as an HTTP request message to the Service Endpoint. Developers can get RDP APIs the Service Endpoint, HTTP operations, and parameters from RDP API Playground page - which is an interactive documentation site developers can access once they have a valid RDP account.

Requesting CFS Data

Step 2: Listing the Package Ids using the Bucket Name

To request the CFS data, the first step is to send an HTTP `GET` request to the RDP `/file-store/v1/packages?bucketName={bucket-name}` endpoint to list all Package Ids under the input `bucket-name`.

The HTTP Request structure is as follows:

```
GET /file-store/v1/packages?bucketName={bucket-name} HTTP/1.1
Host: api.refinitiv.com
Authorization: Bearer <Access Token>
```

The example bucket names and package Ids for RDP content set are as follows:

Content	Bucket Name	Example of Package ID
Financial Markets Reference Data	bulk-FMRD	4d48-d7ff-edcc3d38-8243-a4f7517962b8

Content	Bucket Name	Example of Package ID
Symbology	bulk-Symbology	4c80-73a0-fcef949b-bfde-2b9b8117cfb0
ESG	bulk-ESG	4288-ebb6-93372235-acb2-89882a826af1
ESG - Point in Time	bulk-ESG	4173-aec7-8a0b0ac9-96f9-48e83ddbd2ad
Tick History	TICKHISTORY_VBD_NO_EMBARGO	4c01-ab9e-db594a31-a8f5-5b7852ec4638
Green Revenue	bulk-GreenRevenue	Summary: 4e94-6d63-fea034dc-90e2-de33895bd4e9
Green Revenue	bulk-GreenRevenue	Standard: 4316-d43b-81c40763-8e6a-0dbec8162ab1
Starmine	STARMINE_PREDICTIVE_ANALYTICS_SMARTECON_LIVE	40d4-1404-58533484-afe8-718650a4e0d4
Municipal Market Monitor (TM3)	bulk-Custom	4c5b-0f65-c57f5ae4-b548-2a1075ee725d

Note: The bucket name is *case-insensitive*.

If you cannot find the bucket name for your interested content set, please contact your LSEG representative.

Next, set a bucket name to a `bucket_name` variable below like to following statement:

```
bucket_name = 'bulk-Symbology'
```
This notebook uses bulk-ESG as an example.

In [4]:
```
# set Bucket Name, this notebook use bulk-ESG as an example

bucket_name = 'bulk-ESG'
```

In [ ]:
```
#step 2 - list Package IDs from bucket name

CFS_url = f'{RDP_HOST}/file-store/v1/packages?bucketName={bucket_name}'

try:
    packageID_response = requests.get(CFS_url, headers={'Authorization': f'Bearer {
except requests.exceptions.RequestException as exp:
    print(f'Caught exception: {exp}')

if packageID_response.status_code == 200:  # HTTP Status 'OK'
    print('Receive list Package IDs from RDP APIs')
else:
```

```
    print(f'RDP APIs: CFS request failure: {packageID_response.status_code} {packag
    print(f'Text: {packageID_response.text}')
```

```
Receive list Package IDs from RDP APIs
```

Example of the first entry of package IDs, the pacakgeId is the `packageId` field.

In [ ]:
```
print(json.dumps(packageID_response.json()['value'][0], sort_keys=True, indent=2, s
```


```
{
  "bucketNames":[
    "bulk-ESG"
  ],
  "contactEmail":"robin.fielder@refinitiv.com",
  "created":"2021-11-11T07:54:04Z",
  "modified":"2025-04-04T09:55:15Z",
  "packageId":"4037-e79c-96b73648-a42a-6b65ef8ccbd1",
  "packageName":"Bulk-ESG-Global-Symbology-Organization-v1",
  "packageType":"bulk"
}
```


More on /file-store/v1/packages parameters

Beside the `packageName` query, the `/file-store/v1/packages` endpoint supports the following optional parameters:

- *packageType*: Return all packages that match the specified package type.
- *bucketName*:Return all packages that are associated with the specified bucket name.
- *page*: Filter results by a specific pagination index (If client has already specified this query parameter, the skipToken logic will be excluded)
- *includedTotalResult*: The total search result will be counting and added to the first response message.
- *skipToken*: A token to retrieve the next set of result that exceeds page size.
- *pageSize*: The number of packages that will be shown on one page. Default value is 25.
- *includedEntitilementResult*: CFS will perform a permission check on each package against the client permission.

Please find more detail on the CFS API User Guide document.

Tips

It is recommended to call the endpoint with `pageSize=100` parameter as follows:

```
GET /file-store/v1/packages?bucketName={bucket-name}&pageSize=100 HTTP/1.1
Host: api.refinitiv.com
Authorization: Bearer <Access Token>
```

Step 3: Listing the FileSets using the Bucket Name and Package ID

Now we come to getting the FileSets information. The application needs to send an HTTP `GET` request to the RDP `/file-store/v1/file-sets?bucket={bucket-name}&packageId={packageId}` endpoint to list all FileSets under the input `bucket-name` and `packageId`.

The HTTP Request structure is as follows:

```
GET /file-store/v1/file-sets?bucket={bucket-name}&packageId={packageId}
HTTP/1.1
Host: api.refinitiv.com
Authorization: Bearer <Access Token>
```

Next, set a package Id to `packageId` variable below like to following statement:

```
packageId = '408c-ba29-cae68349-abb7-ba38f6d7fce7'
```

This notebook uses *4037-e79c-96b73648-a42a-6b65ef8ccbd1* packageId of the *bulk-ESG* bucket as an example.

If you cannot find the package Id for your interested content set, please contact your LSEG representative.

In [14]:
```
# pick the packageId you need and set to the packageId variable
#packageId = response.json()['value'][0]['packageId']
packageId = '408c-ba29-cae68349-abb7-ba38f6d7fce7'
```

In [ ]:
```
#step 3 - list FileSets from bucket name and package Id

CFS_url = f'{RDP_HOST}/file-store/v1/file-sets?bucket={bucket_name}&packageId={pack

try:
    fileSet_response = requests.get(CFS_url, headers={'Authorization': f'Bearer {ac
except requests.exceptions.RequestException as exp:
    print(f'Caught exception: {exp}')

if fileSet_response.status_code == 200:  # HTTP Status 'OK'
    print('Receive FileSets list from RDP APIs')
else:
    print(f'RDP APIs: CFS request failure: {fileSet_response.status_code} {fileSet_
    print(f'Text: {fileSet_response.text}')
```

```
Receive FileSets list from RDP APIs
```

Example of the first entry of FileSets.

In [ ]:
```
print(json.dumps(fileSet_response.json()['value'][0], sort_keys=True, indent=2, sep
```

```
{
  "attributes":[
    {
      "name":"ContentType",
      "value":"ESGWealthStandard"
    },
    {
      "name":"ResultCount",
      "value":"3855140"
    }
  ],
  "availableFrom":"2025-06-08T19:25:14Z",
  "availableTo":"2025-06-22T19:25:13Z",
  "bucketName":"bulk-ESG",
  "contentFrom":"2025-06-01T16:20:00Z",
  "contentTo":"2025-06-08T18:08:25Z",
  "created":"2025-06-08T19:25:14Z",
  "files":[
    "412d-ab1a-f20f60b8-8279-b0bc7f78a804"
  ],
  "id":"4035-af6c-bcf9f923-834d-c845673b9608",
  "modified":"2025-06-08T19:25:51Z",
  "name":"Bulk-ESG-Global-Raw-Wealth-Standard-v2-ValueScores-Csv-Delta-2025-06-08T1
8:15:15.433Z",
  "numFiles":1,
  "packageId":"408c-ba29-cae68349-abb7-ba38f6d7fce7",
  "status":"READY"
}
```


The File ID is in the files array.

In [ ]:
```
# try just one file
file_id = fileSet_response.json()['value'][0]['files'][0]
print(file_id)
```

```
412d-ab1a-f20f60b8-8279-b0bc7f78a804
```

More on /file-store/v1/file-sets?bucket parameters

Beside the `bucket` and `packageId` queries, the `/file-store/v1/file-sets?bucket` endpoint supports the following optional parameters:

- *name*: The name of the file-set. Only exactly matched results are returned.
- *packageId*: Package ID
- *status*: Filter file-set by status (Ready/Pending)
- *availableFrom*: Return all file-sets that become visible to permissioned users after the specified Datetime.
- *availableTo*: Return all file-sets that is no longer visible to permissioned user after the specified Datetime.
- *contentFrom*: Filter results by the age of the content within the file-set.
- *contentTo*: Filter results by the age of the content within the file-set.
- *createdSince*: Return all file-sets that have a created date after the specified Datetime

- *modifiedSince*: Return all file-sets that have a modified date after the specified Datetime.
- *attributes*: Return a list of publisher-defined attributes of the file-sets.
- *pageSize*: The number of file-sets that will be shown on one page. Default value is 25.
- *skipToken*: A token to retrieve the next set of file-set result that exceeds page size.

The `modifiedSince` parameter can help an application to limit the returned File-Set only for the File-Set that has been modified after a specified time.

Tips

It is recommended to call the endpoint with `pageSize=100` and `modifiedSince` parameters as follows:

```
GET /file-store/v1/file-sets?bucket={bucket-name}&packageId=
{packageId}&modifiedSince={datetime}&pageSize=100 HTTP/1.1
Host: api.refinitiv.com
Authorization: Bearer <Access Token>
```

Example:

```
GET /file-store/v1/file-sets?bucket=bulk-ESG&packageId=4288-ebb6-93372235-
acb2-89882a826af1&pageSize=100&modifiedSince=2022-01-26T00:00:00Z HTTP/1.1
Host: api.refinitiv.com
Authorization: Bearer <Access Token>
```

Please find more detail on the CFS API User Guide document.

Step 3.1: Listing the FileSets using the Bucket Name - Paging

By default, the `/file-store/v1/file-sets` endpoint always returns 25 results per request. You can adjust the number of return results via the `pageSize` query parameter, the maximum number is **100**.

```
GET /file-store/v1/file-sets?bucket={bucket-name}&packageId=
{packageId}&pageSize={number}, HTTP/1.1
Host: api.refinitiv.com
Authorization: Bearer <Access Token>
```

Let's try with `pageSize=2` as an example.

In [ ]:
```
#step 3.5 - list FileSets from bucket name and package Id - with pageSize 2

CFS_url = f'{RDP_HOST}/file-store/v1/file-sets?bucket={bucket_name}&packageId={pack

try:
    fileSet_response = requests.get(CFS_url, headers={'Authorization': f'Bearer {ac
except requests.exceptions.RequestException as exp:
    print(f'Caught exception: {exp}')

if fileSet_response.status_code == 200:  # HTTP Status 'OK'
```

```
    print('Receive list Package IDs from RDP APIs')
else:
    print(f'RDP APIs: CFS request failure: {fileSet_response.status_code} {fileSet_
    print(f'Text: {fileSet_response.text}')
```

```
Receive list Package IDs from RDP APIs
```

In [ ]:
```
print(json.dumps(fileSet_response.json(), sort_keys=True, indent=2, separators=(','
```

```
{
  "@nextLink":"/file-store/v1/file-sets?bucket=bulk-ESG&packageId=408c-ba29-cae68349
-abb7-ba38f6d7fce7&pageSize=2&skipToken=ZmlsZXNldElkPTQwZWMtOGRkYi1kMWY0Y2EyMS04ZTNj
LTIxNTgwMTdmY2RjMA",
  "value":[
    {
      "attributes":[
        {
          "name":"ContentType",
          "value":"ESGWealthStandard"
        },
        {
          "name":"ResultCount",
          "value":"3855140"
        }
      ],
      "availableFrom":"2025-06-08T19:25:14Z",
      "availableTo":"2025-06-22T19:25:13Z",
      "bucketName":"bulk-ESG",
      "contentFrom":"2025-06-01T16:20:00Z",
      "contentTo":"2025-06-08T18:08:25Z",
      "created":"2025-06-08T19:25:14Z",
      "files":[
        "412d-ab1a-f20f60b8-8279-b0bc7f78a804"
      ],
      "id":"4035-af6c-bcf9f923-834d-c845673b9608",
      "modified":"2025-06-08T19:25:51Z",
      "name":"Bulk-ESG-Global-Raw-Wealth-Standard-v2-ValueScores-Csv-Delta-2025-06-0
8T18:15:15.433Z",
      "numFiles":1,
      "packageId":"408c-ba29-cae68349-abb7-ba38f6d7fce7",
      "status":"READY"
    },
    {
      "attributes":[
        {
          "name":"ContentType",
          "value":"ESGWealthStandard"
        },
        {
          "name":"ResultCount",
          "value":"3233481"
        }
      ],
      "availableFrom":"2025-06-16T05:49:41Z",
      "availableTo":"2025-06-30T05:49:40Z",
      "bucketName":"bulk-ESG",
      "contentFrom":"2025-06-08T18:08:25Z",
      "contentTo":"2025-06-16T05:33:46Z",
      "created":"2025-06-16T05:49:41Z",
      "files":[
        "45a2-b47d-46625457-85d2-ae5e68edeb38"
      ],
      "id":"40ec-8ddb-d1f4ca21-8e3c-2158017fcdc0",
      "modified":"2025-06-16T06:08:04Z",
      "name":"Bulk-ESG-Global-Raw-Wealth-Standard-v2-Analytics-Csv-Delta-2025-06-16T
```

```
05:40:35.140Z",
      "numFiles":1,
      "packageId":"408c-ba29-cae68349-abb7-ba38f6d7fce7",
      "status":"READY"
    }
  ]
}
```

Based on the response data above, you see the API returns **2 entries** per request as set via `pageSize=2` parameter.

Step 3.2: Listing the FileSets using the Bucket Name - nextLink

The `@nextLink` node contains the URL for requesting the next page of query as follows:

```
GET {@nextLink URL}, HTTP/1.1
Host: api.refinitiv.com
Authorization: Bearer <Access Token>
```

Example:

In [ ]:
```
# Step 3.2: Listing the FileSets using the Bucket Name - nextLink
if '@nextLink' in fileSet_response.json():
    next_link = fileSet_response.json()['@nextLink']
    #step 3.5 - list Package IDs from bucket name - with pageSize 2 - navigate to n

    CFS_url = f'{RDP_HOST}{next_link}'

    try:
        fileSet_response = requests.get(CFS_url, headers={'Authorization': f'Bearer
    except requests.exceptions.RequestException as exp:
        print(f'Caught exception: {exp}')

    if fileSet_response.status_code == 200:  # HTTP Status 'OK'
        print('Receive list Package IDs from RDP APIs')
    else:
        print(f'RDP APIs: CFS request failure: {fileSet_response.status_code} {file
        print(f'Text: {fileSet_response.text}')
```

```
Receive list Package IDs from RDP APIs
```

In [ ]:
```
print(json.dumps(fileSet_response.json(), sort_keys=True, indent=2, separators=(',',
```

```
{
  "@nextLink":"/file-store/v1/file-sets?bucket=bulk-ESG&skipToken=ZmlsZXNldElkPTQyND
YtNWE1OC1jNzY0MDVlNi1iNTlhLTUzYjE1ZDc2MWMyZA&packageId=408c-ba29-cae68349-abb7-ba38f
6d7fce7&pageSize=2",
  "value":[
    {
      "attributes":[
        {
          "name":"ContentType",
          "value":"ESGWealthStandard"
        }
      ],
      "availableFrom":"2025-06-16T06:15:53Z",
      "availableTo":"2025-06-30T06:15:52Z",
      "bucketName":"bulk-ESG",
      "contentFrom":"1970-01-01T00:00:00Z",
      "contentTo":"2025-06-16T05:33:46Z",
      "created":"2025-06-16T06:15:53Z",
      "files":[
        "4f9f-6dbb-21420ee2-9a21-883de05afc17"
      ],
      "id":"4108-8540-e334597d-a70d-d9f73f671d1c",
      "modified":"2025-06-16T06:17:07Z",
      "name":"Bulk-ESG-Global-Raw-Wealth-Standard-v2-ValueScores-Jsonl-Init-2025-06-
16T05:40:36.092Z",
      "numFiles":1,
      "packageId":"408c-ba29-cae68349-abb7-ba38f6d7fce7",
      "status":"READY"
    },
    {
      "attributes":[
        {
          "name":"ContentType",
          "value":"ESGWealthStandard"
        }
      ],
      "availableFrom":"2025-06-08T19:05:47Z",
      "availableTo":"2025-06-22T19:05:47Z",
      "bucketName":"bulk-ESG",
      "contentFrom":"2025-06-01T16:20:00Z",
      "contentTo":"2025-06-08T18:08:25Z",
      "created":"2025-06-08T19:05:47Z",
      "files":[
        "4676-87b4-d50b35e7-8136-1bd3b2375774",
        "4706-26a3-3048d56e-b2ed-fda73611ba0b",
        "47c1-bb21-e0928cc4-9b0b-bb0053a68289"
      ],
      "id":"4246-5a58-c76405e6-b59a-53b15d761c2d",
      "modified":"2025-06-08T19:25:50Z",
      "name":"Bulk-ESG-Global-Raw-Wealth-Standard-v2-DataItems-Jsonl-Delta-2025-06-0
8T18:15:15.433Z",
      "numFiles":3,
      "packageId":"408c-ba29-cae68349-abb7-ba38f6d7fce7",
      "status":"READY"
    }
```

```
  ]
}
```

Then you can continue to send requests to URL in `@nextLink` node to get the next page results.

Note: The `/file-store/v1/packages?bucketName={bucket-name}` endpoint on the **step 2** above also supports the Paging feature with the same `pageSize` query parameter and `@nextLink` node too.

Step 4: Get the file URL on AWS S3

The next step is getting the file URL on Amazon AWS S3 service with the RDP `/file-store/v1/files/{file ID}/stream` endpoint.

The HTTP Request structure is as follows:

```
GET /file-store/v1/files/{fileId}/stream?doNotRedirect=true HTTP/1.1
Host: api.refinitiv.com
Authorization: Bearer <Access Token>
```

In [ ]:
```
#step 4 - get file URL from file id

FileID_url = f'{RDP_HOST}/file-store/v1/files/{file_id}/stream?doNotRedirect=true'

try:
    fileID_response = requests.get(FileID_url, headers={'Authorization': f'Bearer {
except requests.exceptions.RequestException as exp:
    print(f'Caught exception: {exp}')

if fileID_response.status_code == 200:  # HTTP Status 'OK'
    print('Receive File URL from RDP APIs')
else:
    print(f'RDP APIs: CFS request failure: {fileID_response.status_code} {fileID_re
    print(f'Text: {fileID_response.text}')
```

```
Receive File URL from RDP APIs
```

The File URL is in the `url` attribute of the response message.

In [ ]:
```
file_url = fileID_response.json()['url']
print(file_url)
```

https://a206464-bulk-esg.s3.amazonaws.com/Bulk-ESG-Global-Raw-Wealth-Standard-v2/2025/06/08/Bulk-ESG-Global-Raw-Wealth-Standard-v2-ValueScores-Delta-2025-06-08T18%3A15%3A15.433Z.csv.gz?x-request-Id=ce431bcb-2cdc-417a-9e86-7e83af4056b1&x-package-id=408c-ba29-cae68349-abb7-ba38f6d7fce7&x-client-app-id=b4842f3904fb4a1fa18234796368799086c63541&x-file-name=Bulk-ESG-Global-Raw-Wealth-Standard-v2-ValueScores-Delta-2025-06-08T18%3A15%3A15.433Z.csv.gz&x-fileset-id=4035-af6c-bcf9f923-834d-c845673b9608&x-bucket-name=bulk-ESG&x-uuid=GESG1-178570&x-file-Id=412d-ab1a-f20f60b8-8279-b0bc7f78a804&x-fileset-name=Bulk-ESG-Global-Raw-Wealth-Standard-v2-ValueScores-Csv-Delta-2025-06-08T18%3A15%3A15.433Z&x-event-external-name=cfs-claimCheck-download&X-Amz-Security-Token=IQoJb3JpZ2luX2VjEIn%2F%2F%2F%2F%2F%2F%2F%2F%2F%2FwEaCXVzLWVhc3QtMSJHMEUCIQCcvGmFcP8z9NzuAtLbxo7%2BTf%2B2k6x45T97X4TgxJrXXQIgDhErnOPKi08fNhR4RvJi4fZgcGK8c2LUUuweYNydmNcqmgIIchAFGgw2NDIxNTcxODEzMjYiDEOFsCw0CxOGxTTJmyr3ASNFewKXgFz70H4SVgKZjdyBZz8Db8SGwYyASGLRSRSbinapoo2JAP2IM%2BKE2edYb345%2BMe7B2dWYP96HRR3NFdtsT15uKffmkLCSbl6slAyeIU5lHtSYbxAeCqyVRStl2adiJlrug0oyuasqLWKm7GAJAIeGX0YXbGkHW54rHOy%2BgtD1aKl3O1bJjoHUKM7VxAPnvKEigr4JSgyoYE8xeB%2BqMfuAotBzS2leoRwB%2FWhL%2Fou5x4%2FDpI%2BFIWlaTMKY65%2BuVmRj2gGkpaZRjQ1GfI2OeVFNvbqAEjRAs1zejTuldhft8OJFX91o%2FOc%2BxSL7H%2FXVuf1aUkwutHEwgY6nQF%2BO9OEXsXlF7JZGbbbffFmnJGZMRxE9k4SgUN2NlkVZD4QsfjIkKqzXrm5nVd8K80fRAnkg19mzuwSk7j4ExgnTO2ryDiWp4pKUL5o6V8bvnQ1HSP9l%2B4BXGMsSL%2B%2B3nXAuRkFgZNJb7Jf%2Bajru6y%2B5m8NZCShT8WGQMMMVqbKRei2mt51cFeYsSFJQV%2FDlaXfucfEKggHZkdQxtBp&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20250617T083506Z&X-Amz-SignedHeaders=host&X-Amz-Expires=21600&X-Amz-Credential=ASIAZLA4M7GHHDCU3QDW%2F20250617%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Signature=6708a8de9ba17525fce0b96ddd96a46ee7a3d817aa46291845724564d5806653

More on /file-store/v1/files/ parameters

Beside the `file_id` query, the `/file-store/v1/files/` endpoint supports the following optional parameters:

- *createdSince*: Return all files that have a created date after the specified Datetime.
- *modifiedSince*: Return all files that have a modified date after the specified Datetime.
- *pageSize*: The number of files that will be shown on one page. Default value is 25.
- *skipToken*: A token to retrieve the next set of file result that exceeds page size

Please find more detail on the CFS API User Guide document.

Step 5: Downloading the file

Once you got the S3 URL. You can download the bulk file using that URL (**as is**). **Do not alter or make any changes to the URL text string**. It will cause unable to download or signature mismatch error.

Note:

- If you cannot download the file, please wait for a while and then retry download the file from the URL. Please do not flush the download requests.
- The code below set `verify = False` property in a `requests` library call to workaround LSEG's beloved ZScaler blocks a download request message. **Do not** set `verify = False` in a Production.

In [ ]:
```python
#step 5 - Download file
import polling2

try:
    print(f'Downloading File from {file_url} ...')
    bulkFile_response = polling2.poll(lambda: requests.get(file_url, verify= False)
                            step = 10,
                            poll_forever = True,
                            check_success= lambda r: r.status_code == 200)
except requests.exceptions.RequestException as exp:
    print(f'Caught exception: {exp}')
```

```
Downloading File from https://a206464-bulk-esg.s3.amazonaws.com/Bulk-ESG-Global-Raw-
Wealth-Standard-v2/2025/06/08/Bulk-ESG-Global-Raw-Wealth-Standard-v2-ValueScores-Del
ta-2025-06-08T18%3A15%3A15.433Z.csv.gz?x-request-Id=ce431bcb-2cdc-417a-9e86-7e83af40
56b1&x-package-id=408c-ba29-cae68349-abb7-ba38f6d7fce7&x-client-app-id=b4842f3904fb4
a1fa18234796368799086c63541&x-file-name=Bulk-ESG-Global-Raw-Wealth-Standard-v2-Value
Scores-Delta-2025-06-08T18%3A15%3A15.433Z.csv.gz&x-fileset-id=4035-af6c-bcf9f923-834
d-c845673b9608&x-bucket-name=bulk-ESG&x-uuid=GESG1-178570&x-file-Id=412d-ab1a-f20f60
b8-8279-b0bc7f78a804&x-fileset-name=Bulk-ESG-Global-Raw-Wealth-Standard-v2-ValueScor
es-Csv-Delta-2025-06-08T18%3A15%3A15.433Z&x-event-external-name=cfs-claimCheck-downl
oad&X-Amz-Security-Token=IQoJb3JpZ2luX2VjEIn%2F%2F%2F%2F%2F%2F%2F%2F%2F%2FwEaCXVzLWV
hc3QtMSJHMEUCIQCcvGmFcP8z9NzuAtLbxo7%2BTf%2B2k6x45T97X4TgxJrXXQIgDhErnOPKi08fNhR4RvJ
i4fZgcGK8c2LUUuweYNydmNcqmgIIchAFGgw2NDIxNTcxODEzMjYiDEOFsCw0CxOGxTTJmyr3ASNFewKXgFz
70H4SVgKZjdyBZz8Db8SGwYyASGLRSRSbinapoo2JAP2IM%2BKE2edYb345%2BMe7B2dWYP96HRR3NFdtsT1
5uKffmkLCSbl6slAyeIU5lHtSYbxAeCqyVRStl2adiJlrug0oyuasqLWKm7GAJAIeGX0YXbGkHW54rHOy%2B
gtD1aKl3O1bJjoHUKM7VxAPnvKEigr4JSgyoYE8xeB%2BqMfuAotBzS2leoRwB%2FWhL%2Fou5x4%2FDpI%2
BFIWlaTMKY65%2BuVmRj2gGkpaZRjQ1GfI2OeVFNvbqAEjRAs1zejTuldhft8OJFX91o%2FOc%2BxSL7H%2F
XVuf1aUkwutHEwgY6nQF%2BO9OEXsXlF7JZGbbbffFmnJGZMRxE9k4SgUN2NlkVZD4QsfjIkKqzXrm5nVd8K
80fRAnkg19mzuwSk7j4ExgnTO2ryDiWp4pKUL5o6V8bvnQ1HSP9l%2B4BXGMsSL%2B%2B3nXAuRkFgZNJb7J
f%2Bajru6y%2B5m8NZCShT8WGQMMMVqbKRei2mt51cFeYsSFJQV%2FDlaXfucfEKggHZkdQxtBp&X-Amz-Al
gorithm=AWS4-HMAC-SHA256&X-Amz-Date=20250617T083506Z&X-Amz-SignedHeaders=host&X-Amz-
Expires=21600&X-Amz-Credential=ASIAZLA4M7GHHDCU3QDW%2F20250617%2Fus-east-1%2Fs3%2Faw
s4_request&X-Amz-Signature=6708a8de9ba17525fce0b96ddd96a46ee7a3d817aa46291845724564d
5806653 ...
```

```
c:\Projects\Code\CFS_demo\run_cfs\Lib\site-packages\urllib3\connectionpool.py:1097:
InsecureRequestWarning: Unverified HTTPS request is being made to host 'a206464-bulk
-esg.s3.amazonaws.com'. Adding certificate verification is strongly advised. See: ht
tps://urllib3.readthedocs.io/en/latest/advanced-usage.html#tls-warnings
  warnings.warn(
```

Now you have downloaded the CFS bulk file stream in an application level. You can choose to save that file with whatever name you want.

If you need an actual file name of the file, it is available in S3 URL as follows:

```
https://XXXX.s3.amazonaws.com/XXX/YEAR/MONTH/DATE/{file_name}?x-
request-Id={signature}.
```

Examples:

- https://a206464-bulk-esg.s3.amazonaws.com/Bulk-ESG-Global-Symbology-Organization-v1/2023/11/26/*Bulk-ESG-Global-Symbology-Organization-v1-Init-2023-

11-26T16%3A04%3A11.525Z.jsonl.gz*?x-request-Id=signature (an actual file name is **Bulk-ESG-Global-Symbology-Organization-v1-Init-2023-11-26T16_04_11.525Z.jsonl.gz**)

- https://a206464-bulk-custom.s3.amazonaws.com/GE-11328/2025/06/12/*TM3_SIFMAIndex2025-06-12T14%3A00%3A00.000-04%3A00*?x-request-Id=signature (an actual file name is **TM3_SIFMAIndex2025-06-12T14_00_00.000-04_00**)

The actual file name has been replace a `_` (underscore) with `%3A` escape character, so an application needs to replace the escape character `%3A` with `_` (underscore) character to get an actual file name.

In [ ]:
```
# Save the file locally.
if bulkFile_response.status_code == 200:  # HTTP Status 'OK'
    zipfilename = file_url.split("?")[0].split("/")[-1].replace("%3A","_")
    print('Download File Successfully')
    open(zipfilename, 'wb').write(bulkFile_response.content)
    print(f'{zipfilename} Saved')
else:
    print(f'RDP APIs: Request file failure: {bulkFile_response.status_code} {bulkFi
    print(f'Text: {bulkFile_response.text}')
```

```
Download File Successfully
Bulk-ESG-Global-Raw-Wealth-Standard-v2-ValueScores-Delta-2025-06-08T18_15_15.433Z.cs
v.gz Saved
```

Now you get the CFS file that you can extract and read the file.

That is all for the RDP CFS File workflow.

Step 6: Refresh Token with RDP APIs

Before the session expires (based on the `expires_in` parameter, in seconds) , an application needs to send a Refresh Grant request message to RDP Authentication service to get a new access token before further request data from the platform.

The API requires the following access credential information:

- Refresh Token: The current Refresh Token value from the previous RDP Authentication call
- Client ID: This is also known as `AppKey` , and it is generated using an App key Generator. This unique identifier is defined for the user or application and is deemed confidential (not shared between users). The client_id parameter can be passed in the request body or as an “Authorization” request header that is encoded as base64.
- Grant Type `refresh_token` : This is for getting a new Access Token.

The HTTP request for the RDP APIs Authentication service is as follows:

```
POST /auth/oauth2/v1/token HTTP/1.1
Accept: */*
Content-Type: application/x-www-form-urlencoded
Host: api.refinitiv.com:443
Content-Length: XXX

refresh_token={current_refresh_token}
&grant_type=refresh_token
&client_id=RDP_APP_KEY
```

Once the authentication success, the function gets **access_token**, **refresh_token**, and **expires_in** from the RDP Auth service response message the same as the previous RDP Authentication call. An application must keep those value for the next Refresh Token call.

Caution: API Limit

The RDP Authentication service has the API limit described on the RDP APIs: Limitations and Guidelines for the RDP Authentication Service article. If the application flushes the authentication request messages (both `password` and `refresh_token` grant_type) beyond the limit, the account will be blocked by the API Gateway.

In [ ]:
```
#step 6 - Refreshing Token

# Send HTTP Request
auth_url = f'{RDP_HOST}/auth/oauth2/v1/token'
payload = f'grant_type=refresh_token&client_id={clientId}&refresh_token={refresh_to
auth_response = None

try:
    auth_response = requests.post(auth_url,
                                  headers = {'Content-Type':'application/x-www-form-urle
                                  data = payload,
                                  auth = (clientId, '')
                    )
except requests.exceptions.RequestException as exp:
    print(f'Caught exception: {exp}')

if auth_response.status_code == 200:  # HTTP Status 'OK'
    print('Refresh Token success')
    access_token = auth_response.json()['access_token']
    refresh_token = auth_response.json()['refresh_token']
    expires_in = int(auth_response.json()['expires_in'])

if auth_response.status_code != 200:
    print(f'RDP authentication failure: {auth_response.status_code} {auth_response.
    print(f'Text: {auth_response.text}')
```

```
Refresh Token success
```

Step 7: Revoke Token to ending the session.

This revocation mechanism allows an application to invalidate its tokens if the end-user logs out, changes identity, or exits the respective application. Notifying the authorization server

that the token is no longer needed allows the authorization server to clean up data associated with that token (e.g., session data) and the underlying authorization grant.

The API requires the following HTTP Header and Credential parameter information:

- Header:
  - Authorization = `Basic <App Key in Base64 format>`

Please notice *the space* between the `Basic` and `App Key in Base64 format` values.

- Body parameter
  - token: The current `Access Token` value from the previous RDP Authentication call

The HTTP request for the RDP APIs Authentication service is as follows:

```
POST /auth/oauth2/v1/revoke HTTP/1.1
Accept: */*
Content-Type: application/x-www-form-urlencoded
Host: api.refinitiv.com:443
Authorization: Basic <App Key in Base64>
Content-Length: XXX

token={current_Access_token}
```

In [ ]:

```
#step 7 - Revoking Token

import base64

clientId_bytes = clientId.encode('ascii')
base64_bytes = base64.b64encode(clientId_bytes)
clientId_base64 = base64_bytes.decode('ascii')

# Send HTTP Request
auth_url = f'{RDP_HOST}/auth/oauth2/v1/revoke'
payload = f'token={access_token}'
auth_response = None

try:
    auth_response = requests.post(auth_url,
                                  headers = {
                                      'Content-Type':'application/x-www-form-urlencoded'
                                      'Authorization': f'Basic {clientId_base64}'
                                  },
                                  data = payload,
                                  auth = (clientId, '')
                    )
except requests.exceptions.RequestException as exp:
    print(f'Caught exception: {exp}')

if auth_response.status_code == 200:  # HTTP Status 'OK'
    print('Revoke Token success')
if auth_response.status_code != 200:
```

```
    print(f'RDP authentication failure: {auth_response.status_code} {auth_response.
    print(f'Text: {auth_response.text}')
```

```
Revoke Token success
```

That's all I have to say about the CFS API workflow.

Next Steps

You may interested in the following resources for more detail about the CFS data usage:

- Find environmental footprint of your bond portfolio article
- RDP APIs Green Revenues CFS file Workflow - a dedicate Green Revenue CFS workflow
- RDP APIs ESG CFS file Workflow - a dedicate ESG CFS workflow

And much more on the Developer Portal website.

References

That brings me to the end of my generic CFS file workflow project. For further details, please check out the following resources:

- RDP APIs page on the LSEG Developer Community website.
- RDP APIs Playground page.
- RDP APIs: Introduction to the Request-Response API.
- RDP APIs: Authorization - All about tokens.
- Limitations and Guidelines for the RDP Authentication Service article.
- Getting Started with Data Platform article.
- CFS API User Guide.

For any questions related to RDP APIs or CFS service, please use the Developers Community Q&A page.

In [ ]: