Oracle Financial Services
File Upload and Download Utility User Guide

Release 23.03.01

F78535-09

March 2023

ORACLE®

Oracle Financial Services File Upload and Download Utility User Guide, Release 23.03.01

F78535-09

Contents

ORACLE®

1

Get Help

Topics:

Get Help in the Applications

Use help icons to access help in the application.

Note that not all pages have help icons. You can also access the Oracle Help Center to find guides and videos.

Additional Resources

- Community: Use Oracle Cloud Customer Connect to get information from experts at Oracle, the partner community, and other users.
- Training: Take courses on Oracle Cloud from Oracle University.

Learn About Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program. Videos included in this guide are provided as a media alternative for text-based topics, and are also available in this guide.

Get Support

You can get support at My Oracle Support.

For accessibility support, visit Oracle Accessibility Learning and Support.

Get Training

Increase your knowledge of Oracle Cloud by taking courses at Oracle University.

Join Our Community

Use Cloud Customer Connect to get information from industry experts at Oracle and in the partner community. You can join forums to connect with other customers, post questions, and watch events.

Share Your Feedback

We welcome your feedback about Oracle Applications user assistance. If you need clarification, find an error, or just want to tell us what you found helpful, we would like to hear from you.

You can email your feedback to My Oracle Support.

Thanks for helping us improve our user assistance!

Before You Begin

See the following Document:

• See What's New

2
Roles and Functions

The following table lists the role codes and function codes required to configure the File Upload/Download Utility.

Role Code	Function Code
FILE_READ	FILE_SUMMARY
FILE_UPLOAD	FILE_UPLOAD
FILE_DOWNLOAD	FILE_DOWNLOAD
FILE_ADV	FILE_UPLOAD
	FILE_DOWNLOAD
	FILE_DELETE
	FILE_SUMMARY

3
File Upload and Download Utility

The File Upload and Download Utility enables you to upload or download files to the Object Store. Complete the following steps to Upload or Download a file.

Upload or Download File from Object Store Using Console

1. From the left menu, click **Common Object Maintenance**.
2. Click **Data Management** in the left navigation pane.

   The **File Upload and Download** Page is displayed. The Files that are uploaded to the Object Store are listed here. The following details are provided for each File.

Field	Description
File ID	The unique file ID associated with the file. This is auto-generated during the upload.
Prefix	The prefix is added to the file name.
File Name	The name of the file that is uploaded. This is automatically updated after you select the file.
Stripe Name	The Unique Identifier for storing a collection of files. Collection examples: Project, organization, tenant.
Uploaded Date	The file upload date.
Download File	Click **Download** to download a copy of the uploaded file.
Delete	Click **Delete** to delete the file.

Related Topics

- Uploading/Downloading a File Using Utility
  Complete the following steps to Upload or Download a file using the Utility.

- Uploading/Downloading a File Using PAR URL
  Complete the following steps to upload or download a file using the PAR URL.

Uploading/Downloading a File Using Utility

Complete the following steps to Upload or Download a file using the Utility.
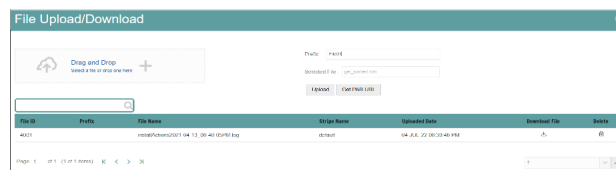
1. Click **Drag and Drop** to browse and select a file for upload from the local directory.

You can also browse to the local directory from the **File Explorer** and select file and drop it here.

The file name is automatically updated in the **Selected File** field.

2. Enter the **Prefix** to be added to the file name.

The Prefix is added to the file name. In case, you have two files with the same file name, you can save then with different prefixes.
Example: **/abc/test.txt** and **/def/test.txt**. Both these files have the same file name but different Prefixes.

3. Click **Upload** to upload the selected file.

A confirmation message is displayed after successful upload and the file is listed in the Uploaded Files list.

Uploading/Downloading a File Using PAR URL

Complete the following steps to upload or download a file using the PAR URL.

Figure 3-1   Get PAR URL
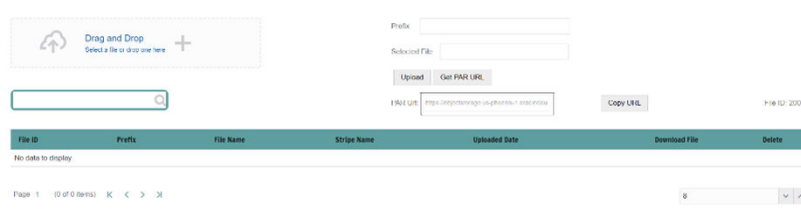
1. Click **Drag and Drop** to browse and select a file for upload from the local directory.

You can also browse to the local directory from the File Explorer and select file and drop it here. The file name is automatically updated in the **Selected File** field.

2. Enter the **Prefix** to be added to the file name.

3. Click **Get PAR URL**.

This will generate the PAR URL and File ID which are required in order to upload the file.

You can also generate PAR URL using Rest API. For more information refer to, Calling the API to Generate the URL .

4. Copy PAR URL and note the related File ID.

5. You can upload file content referred with the specific PAR URL into the object store using one of the following options: Console, CLI, or SDK.

For example, you may use the following curl command directly in local Gitbash.

```
curl -X PUT --data-binary '@<local-filename>' <unique-PAR-URL>
```

You can also use the following command.

```
curl -T '<Filepath>' -X PUT <PAR_URL>
```

You can scan the file referred with the specific File ID (obtained in Step 3) using one of the following options: Console, CLI, or SDK. For example, use the following CURL command to scan the File. Use the File ID.

```
curl -k --location --request PUT 'https://<Host:Port>/<Tenant-ID>/utils-
service/v1/file/scan/<FileID>' \
--header 'ofs_tenant_id: <Tenant-ID>' \
--header 'ofs_service_id: <Service-ID>' \
--header 'ofs_workspace_id: <WorkspaceId>' \
--header 'Authorization: Bearer <Generated_Token>'
```

A confirmation message is displayed after successful upload and the file is listed in the Uploaded Files list.

4

File Upload Automation

To simplify the file upload process, configure and execute the File Upload automation utility.

File Upload automation can be implemented by completing the following steps in sequence.

- Background Information
- Step 1: Generate Access Token
- Step 2: Generate PAR URL
- Step 3: Upload file to Object Store
- Step 4: Scan the file to ensure Upload was Successful

Background Information

Client for URLs

For information regarding how to use Client for URLs (cURL), access the following link:

Sending API requests using cURL

View list of Uploaded Files

For information regarding how to view a list of uploaded files in your Profitability and Balance Sheet Management Cloud Service, access the following link:

Viewing List of Uploaded Files

Step 1: Generate Access Token

Generate the Access Token for your Profitability and Balance Sheet Management Cloud Service by:

1. Submitting a RESTful API Post Request to your Oracle IDCS environment as defined in the Identity Cloud Service User Guide. For information, see OAuth Runtime Tokens REST Endpoints.

Note that a sample code snippet has been provided below using cURL to generate the access token for Basic Authorization and assign it to a variable for use within a script:

```
access_token=`curl -s --insecure -H "Authorization: Basic $ENCODED" -H
"Content-Type: application/x-www-form-urlencoded;charset=UTF-8" --request
POST $IDCS_URL -d
"grant_type=password&username=$USERNAME&password=$PASSWORD&scope=urn:opc:idm:
__myscopes__ urn:opc:resource:expiry=9153600" | python3 -c "import sys,
json; print(json.load(sys.stdin)['access_token'])"`
```

Step 2: Generate PAR URL

Generate the PAR URL your Profitability and Balance Sheet Management Cloud
Service by:

1. Submitting a RESTful API Post Request to your Profitability and Balance Sheet
    Management Cloud Service as defined in the Calling the API to Generate the URL
    section.

Note a sample code snippet has been provided below using cURL to generate the
PAR URL and and assign it to a variable for use within a script:

PAR URL Generation Code Snippet

```
curl --location --insecure --request POST "$FILEUPLOADURL" --header
"Authorization: Bearer $access_token" --header 'Content-Type:
application/json' --data-raw "{
    \"fileName\": \"$1\",
    \"fileSize\": \"$2\",
    \"mimeType\": \"$3\"
}" >> "$HOME"/FILEUPLOAD_UTIL/"$1"_PARURLresponse.out 2>&1
```

PAR URL Variable Assignment Code Snippet

```
grep -oE '(https)[^ ]*' "$HOME"/
FILEUPLOAD_UTIL/"$1"_PARURLresponse.out > "$HOME"/
FILEUPLOAD_UTIL/"$1"_PARURLresponse1.csv
    while IFS="," read -r H1 H2
    do
        URLtrim="$H1"
        FinalPAR=${URLtrim::-1}
          echo -e "\n PAR_URL is $FinalPAR"
    done < "$HOME"/FILEUPLOAD_UTIL/"$1"_PARURLresponse1.csv
```

PAR URL Variable Assignment Code Snippet – used in File Scanning Step

```
grep -oE '(fileId)[^ ]*' "$HOME"/
FILEUPLOAD_UTIL/"$1"_PARURLresponse.out > "$HOME"/
FILEUPLOAD_UTIL/"$1"_PARURLresponse2.csv
    while IFS="," read -r F1 F2
    do
        FILEIDtrim="$F1"
        FINALFILEID=`echo "$FILEIDtrim"| sed -r 's/^.{8}//'`

        echo -e "\n FILE ID is $FINALFILEID"
    done < "$HOME"/FILEUPLOAD_UTIL/"$1"_PARURLresponse2.csv
```

Step 3: Upload file to Object Store

Upload file to the Object Store of your Profitability and Balance Sheet Management Cloud Service by:

1. Submitting a RESTful API Post Request to your Profitability and Balance Sheet Management Cloud Service as defined in the Uploading/Downloading a File Using PAR URL topic.

Note a sample code snippet has been provided below using cURL to upload the file:

```
curl -T "$HOME"/FILEUPLOAD_UTIL/"$1" -X PUT "$FinalPAR"
```

Step 4: Scan the file to ensure Upload was Successful

Scan the file that was uploaded to the Object Store of your Profitability and Balance Sheet Management Cloud Service by:

1. Submitting a RESTful API Put Request to your Profitability and Balance Sheet Management Cloud Service as defined by the code snippet below:

File Scanning Code Snippet – using File ID from Step 2 – Generate PAR URL

```
if [ $last_error -eq 0 ]; then
        echo -e "\n *****Scanning the File *****"
        curl -k --location --request PUT "$SCANURL/$FINALFILEID" --header
'ofs_tenant_id: $TENANT' --header 'ofs_service_id: OFS_FTP' --header
'ofs_workspace_id: WS001' --header "Authorization: Bearer $access_token"
        last_error=$?
        if [ $last_error -eq 0 ]; then
            echo -e "\n ***File Upload is Successful please check File
Upload / Download UI***"
        else
            echo "Scan failed"
            exit -2;
        fi
    else
        echo "Upload failed"
        exit -3;
    fi
```

Automating the File Upload Process Using File Upload Utility

This section provides the procedure including the script to automate the process of uploading input data files using the File Upload utility.

You can download this script from this MoS Doc ID 2927077.1.

Executing the File Upload Automation Script

File Upload Automation script assists you to upload the files seamlessly.

Complete the following procedure to execute the fileupload automation script.

Python 3.10 is required to access data elements from the API JSON responses.

1. Extract the **FILEUPLOAD_UTIL.zip** file located in the **$HOME** directory.

2. Copy the Data Loader input file to the **$HOME/FILEUPLOAD_UTIL** directory.

Figure 4-1   Data Loader Input File Location Path

3. Edit the script **Env_setup.sh** file to update the following environment variables.

   • **IDCS URL** - The Service Instance URL to access your IDCS console.
     You can get the IDCS URL from the following menu:

     Oracle Cloud Infrastructure Console > Identity Cloud.

     Syntax: `<idcs-url>/oauth2/v1/token`

   • **Encoded** - The **<OAuth Client ID>:<OAuth Client Secret>** encoded using **base64encode**
     To extract and encode the Client ID and Client Secret, refer to the following steps:

     a. Login to **Admin Console**.

     b. Go to **System Configuration** tab, and click **Component Details** tile.

     c. Click **OAUTH Creds** tab to view and copy the **OAUTH Client ID** and **OAUTH Client Secret** details.

     d. Using any **base64encode** utility, encode <OAUTH Client ID>:<OAUTH Client Secret>.
        Example (Input OAUTH Client ID and Client Secret):

        `ftptenant-prd_APPID:99140e14-4d30-4e86-85fb-09501fe45fe0`

        Example (Encoded OAUTH Client ID and Client Secret):
        `ZnRwcWExMDEyMzEtcHJkX0FQUElEOjBkMmU5MDBiLTlhYjItNGFmOS05OWM0LTE wNTYyMDVkYWYwNQ==`

   • **Username** - The Username to access the application.

> **Note:**
>
> The user should have appropriate roles and privileges. For more information about roles and priveleges, refer to Roles and Functions.

- **Password** - The password to login to the application.

- **Tenant** - The tenant associated with the application.

- **PBSM Host** - Details of the PBSM host on which the application is hosted.
Sample Env_setup.sh

```
IDCS_URL=https://idcs-xyz123.identity.c9xyz.oc9xyz.com/oauth2/v1/token
ENCODED=ZnRwbWFydXAxNDIyMzEtcHJkX0FQUElEOjk5MTQwwZTE0LTRkMzAtNGU4Ni04NW
ZiLTA5NTAxZmU0NWZlMA==
USERNAME=<user_name>
PASSWORD=<password>
TENANT=<tenant-prd>
PBSMHOST=dc.pbsmcloud.us-xxxxx -1.ocs.oc-test.com
```

4. Execute **PBSMCS_PAR_fileupload.sh** with the following parameters –

- **filename** - The file to be uploaded

- **filesize** - The file size in Bytes

- **Mimetype** - The mimetype of the file.

Format: `Filename<space>Filesize<space>mimetype`

Example: `./PBSMCS_PAR_fileupload.sh input_20150101_filename_example_8007.txt 334 text/plain`

This script generates the access token and the PAR URL. It also uploads the file into the object store and scans the file too.

> **Note:**
>
> To upload multiple files, you must execute the script for each file, separately.

Once the script is executed succesfully, the file is uploaded and added to the list of files in the File Upload/Download page. To access the File Upload/Download page refer to Upload or Download File from Object Store Using Console.

5
Generating PAR URL for File Operations

The PAR URL for File Operations API creates a PAR File that you can use to perform file operations in the Object Store for end-to-end integrations.

Generating PAR URL for File Upload

You can use this REST API to generate the PAR URL for File Upload. See the following sections for information on how to perform the POST operation.

Related Topics

*   Calling the API to Generate the URL

End Point Details

*   **Method** – POST
*   **URL** - `https://<HOST_NAME:PORT>/<TENANT>/utils-service/v1/file/uploadfile/parURL?prefix=<prefix>`
*   **Content-Type** - Application/Json

Calling the API to Generate the URL

To call the API, follow these steps:

1. Open a relevant tool, such as via cURL command.
2. Prepare a cURL command with the authentication token and other details. For more information refer to the following code.

    Syntax

    ```
    curl -k --location --request POST 'https://<hostname>/<TENANT-ID>/utils-
    service/v1/file/uploadfile/parURL?prefix=' \
    --data-raw '{"fileName": "<remote filename>", "fileSize": <file size>,
    "mimeType": "<file type>"}' \
    --header 'ofs_remote_user: <USERID>' \
    --header 'locale: en-US' \
    --header 'ofs_tenant_id: <TENANT-ID>' \
    --header 'ofs_workspace_id: WS001' \
    --header 'content-type: application/json' \
    --header 'Authorization: Bearer <TOKEN>'
    ```

    Example (truncated)

    ```
    curl -k --location --request POST 'https://<hostname>/<TENANT-ID>/utils-
    service/v1/file/uploadfile/parURL?prefix=' \
    --data-raw '{"fileName": "idcs_log1.txt", "fileSize": 100, "mimeType":
    ```

```
"text/plain"}' \
--header 'ofs_remote_user: cneadmin' \
--header 'locale: en-US' \
--header 'ofs_tenant_id: aaitestdev1001-prd' \
--header 'ofs_workspace_id: WS001' \
--header 'content-type: application/json' \
--header 'Authorization: Bearer
eyJ4NXQjUzI1NiI6Ildia25rQUR5TUZIMlhlQ1pKcTY1c3o4VzdEVWhKa0s4MldYY0ha
dk4wWkkiLCJ4
...
...
sQXj0iohsSIEmQXVwwjhhqnc4eJNnmCjx8Tb7TXjx1MIQLeOIcfrIj5gkzoMKX94_7US
xHv-6LhBzw'
```

Request JSON Parameters

This section provides the list of parameters in the JSON Request.

Table 5-1   Request JSON Parameters

Name	Type	Required	Description
fileName	STRING	Yes	The name of the file to be uploaded. The following are the conditions for to enter in this field: • Must start with an Alphanumeric Character • Allowed characters are alphabets, numbers, and special characters - hyphen(-), dot(.), and underscore(_) • Length of characters must not be greater than 255 characters

Table 5-1   (Cont.) Request JSON Parameters

Name	Type	Required	Description
fileSize	INTEGER	Yes	The size of the file (in Bytes) to be uploaded.
			The size of the file should be greater than 1 Byte and should be less than 10 TB.
			It is recommended to use multipart upload for uploading files of size more than 100 MB. For more information about uploading large objects and multipart uploads, refer to Working with Pre-Authenticated Requests.
mimeType	STRING	Yes	The mime type to be uploaded.
			The following mime types are allowed:
			• Text/CSV
			• Text/plain
			• DAT

Request JSON Sample

```
 [{
"fileName": "File.csv",
"fileSize": 7654,
"mimeType": "text/csv"
}]
```

Response JSON Parameters

This section provides the list of parameters in the JSON Response.

Table 5-2   POST JSON Response

Name	Type	Description
fileName	STRING	The name of the file to be uploaded.
uploadURL	STRING	The generated pre-authenticated URL to upload a file.
fileId	INTEGER	The unique File Identifier.

Response JSON Sample

```
{
    "payload": {
        "uploadURL": "https://objectstorage.us-
phoenix-1.oraclecloud.com/p/bdSI-
hzigiAoUU0lyEKnuk0YGs05L172gt_woZAgqNFYmUFQeexV3BDfTO97mhBI/n/
oraclegbudevcorp/b/fsgbu_pbsm_cndevcorp_ftpqa101231-prd_default/o/
default/2023-01-31/jfr/f9ce031f-4a42-471d-b4da-d0577f3eca15",
        "createUser": "user1",
        "stripeName": "default",
        "fileId": 5025,
        "createDate": "2023-01-31T09:14:16",
        "token": "",
        "status": "success"
    }
}
```

Viewing List of Uploaded Files

Run the following cURL command to generate and view all the files that are uploaded using PAR URL.

Syntax

```
curl -k --location --request GET 'https://<hostname>/<TENANT-ID>/utils-
service/v1/listfiles stripeName=default' \
--header 'locale: en-US' \
--header 'ofs_remote_user: <user id>' \
--header 'ofs_tenant_id: < TENANT-ID >' \
--header 'ofs_workspace_id: WS001' \
--header "Authorization: Bearer <TOKEN>'
```

Example

```
curl -k --location --request GET 'https://dc.pbsmcloud.us-
phoenix-1.ocs.oc-test.com/aaitestdev1001-prd/utils-service/v1/
listfiles?stripeName=default' \
--header 'locale: en-US' \
--header 'ofs_remote_user: cneadmin' \
--header 'ofs_tenant_id: aaitestdev1001-prd' \
--header 'ofs_workspace_id: WS001' \
--header "Authorization: Bearer ${TOKEN}"
```

Response

```
    {"payload":
[{"v_file_name":"Idcs_log3.txt","n_file_id":9916,"d_upload_date":"31-
JAN-23
    06:33:43 AM","v_stripe_name":"default"},
{"v_file_name":"Idcs_log4.txt","n_file_id":9917,"d_upload_date":"31-
```

```
JAN-23
    06:40:25 AM","v_stripe_name":"default"}],"count":2}
```

Generating PAR URL For File Download

You can use this REST API to generate the PAR URL for File Download. See the following sections for information on how to perform the post operation.

Calling the API to Generate PAR URL for File Download Using File Name

To call the API, follow these steps:

1. Open a relevant tool, such as via cURL command.

2. Prepare a cURL command with the authentication token and other details. For more information refer to the following code.

   Syntax

   ```
   curl -k --location --request GET < 'https://<hostname>/<TENANT-ID>/utils-
   service/v1/file/download?fileName=<file name>&stripeName=default&prefix='
   \
    --header 'ofs_remote_user: <userid>' \
    --header 'locale: en-US' \
    --header 'ofs_tenant_id: <TENANT-ID>' \
    --header 'ofs_workspace_id: WS001' \
    --header "Authorization: Bearer <TOKEN>"
   ```

   Example

   ```
   curl -k --location --request GET 'https://<hostname>/<TENANT-ID>/utils-
   service/v1/file/download?fileName=test3GB.xml&stripeName=default&prefix='
   \
    --header 'ofs_remote_user: cneadmin' \
    --header 'locale: en-US' \
    --header 'ofs_tenant_id: aaitestdev1001-prd' \
    --header 'ofs_workspace_id: WS001' \
    --header "Authorization: Bearer ${TOKEN}"
   ```

   Response

   ```
    {"payload":{"downloadURL":"https://objectstorage.us-
   phoenix-1.oraclecloud.com/p/8R68eVcQAxQjNjK__S04MZjS-
   v4BqEbWSILvu0w40kJNrzfKeCB8vWBwugW5XvsK/n/oraclegbudevcorp/b/
   fsgbu_pbsm_cndevcorp_aaitestdev1001-prd_default/o/default/2023-01-20/rnz/
   6c023e75-09e2-4265-815e-32cedcd2415e?
   httpResponseContentDisposition=ATTACHMENT%3B%20filename%3Dtest3GB.xml"}}
   ```

Calling the API to Generate PAR URL for File Download Using File ID

To call the API, follow these steps:

1. Open a relevant tool, such as via cURL command.

2. Prepare a cURL command with the authentication token and other details. For more information, refer to the following code.

Syntax

```
curl -k --location --request GET ' 'https://<hostname>/<TENANT-ID> /
utils-service/v1/file/downloadfile/<file id>' \
--header 'ofs_remote_user: <userid>' \
--header 'locale: en-US' \
--header 'ofs_tenant_id: < TENANT-ID> ' \
--header 'ofs_workspace_id: WS001' \
--header "Authorization: Bearer <TOKEN>"
```

Example

```
curl -k --location --request GET 'https://<hostname>/<TENANT-ID>/
utils-service/v1/file/downloadfile/9916' \
--header 'ofs_remote_user: cneadmin' \
--header 'locale: en-US' \
--header 'ofs_tenant_id: aaitestdev1001-prd' \
--header 'ofs_workspace_id: WS001' \
--header "Authorization: Bearer ${TOKEN}"
```

Response

```
{"payload":{"downloadURL":"https://objectstorage.us-
phoenix-1.oraclecloud.com/p/
gTxxzhqLEea4Or2TRkBqTqHxt_JogVFa9G_0wtN8NYy_op0Zk4lvKGDxxeXGhLq7/n/
oraclegbudevcorp/b/fsgbu_pbsm_cndevcorp_aaitestdev1001-
prd_default/o/default/2023-01-31/fae/2d63d2fe-2090-4fb7-
a4c8-9940d22987db?
httpResponseContentDisposition=ATTACHMENT%3B%20filename%3DIdcs_log3.
txt"}}
```