Oracle Financial Services File Upload and Download Utility User Guide



Release 22.12.01 F77451-01 January 2023

ORACLE

Oracle Financial Services File Upload and Download Utility User Guide, Release 22.12.01

F77451-01

Copyright © 2022, 2023, Oracle and/or its affiliates.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle®, Java, and MySQL are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

1 Get Help

1.1	Get Help in the Applications	1-1
	1.1.1 Additional Resources	1-1
1.2	Learn About Accessibility	1-1
1.3	Get Support	1-1
1.4	Get Training	1-1
1.5	Join Our Community	1-2
1.6	Share Your Feedback	1-2
1.7	Before You Begin	1-2

2 Roles and Functions

3 File Upload and Download Utility

3.1	Upload or Download File from Object Store Using Console	3-1
3.2	Uploading/Downloading a File Using Utility	3-1
3.3	Uploading/Downloading a File Using PAR URL	3-2

4 File Upload Automation

4.1 Fi	e Upload Process Outline	4-1
4.1.	1 Background Information	4-1
4.1.	2 Step 1: Generate Access Token	4-1
4.1.	3 Step 2: Generate PAR URL	4-1
4.1.	4 Step 3: Upload file to Object Store	4-2
4.1.	5 Step 4: Scan the file to ensure Upload was Successful	4-3
4.2 A	itomating the File Upload Process Using File Upload Utility	4-3
4.2.	1 Software Prerequisites	4-3
4.2.	2 File Upload Using File Upload Utility	4-3



5 Generating PAR URL for File Operations

5.1	Gene	erating PAR URL for File Upload	5-1
	5.1.1	End Point Details	5-1
	5.1.2	Calling the API to Generate the URL	5-1
	5.1	.2.1 Request JSON Parameters	5-2
	5.1	.2.2 Response JSON Parameters	5-3
	5.1.3	Viewing List of Uploaded Files	5-4
5.2	Gene	erating PAR URL For File Download	5-4
	5.2.1	Calling the API to Generate PAR URL for File Download Using File Name	5-4
	5.2.2	Calling the API to Generate PAR URL for File Download Using File ID	5-5



1 Get Help

Topics:

- Get Help in the Applications
- Learn About Accessibility
- Get Support
- Get Training
- Join Our Community
- Share Your Feedback
- Before You Begin

1.1 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.

1.1.1 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.

1.2 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.

1.3 Get Support

You can get support at My Oracle Support.

For accessibility support, visit Oracle Accessibility Learning and Support.

1.4 Get Training

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



1.5 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.

1.6 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!

1.7 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.

3.1 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.

3.2 Uploading/Downloading a File Using Utility

Complete the following steps to Upload or Download a file using the Utility.

			Prets. Head						
Drag and Dra Select a tel or org		e ten +	Beledied File : get_started to	Beleded File : get_started him					
			Upload Car PNR URI						
		Q							
18e ID	Prefix	Tile Name	Stripe Name	Uploaded Date	Download File	Delete			
001		retal/chors2021 04 13 05 40 05/M log	dotaut	04 JUL 22 08/32 46 PM	*	6			

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: *labc/test.txt* and *ldef/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.

3.3 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

O INFOR International contractors	Copy URL	H(e ID: 2002
File ID Prefix File Name Stripe Name Uploaded Date	Download File	Delete

- 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

This chapter consists of the following topics:

- File Upload Process Outline
- Automating the File Upload Process Using File Upload Utility

4.1 File Upload Process Outline

This section provides the step-by-step process that is implemented within the File Upload utility.

4.1.1 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

4.1.2 Step 1: Generate Access Token

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

 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'])"`
```

4.1.3 Step 2: Generate PAR URL

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



 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

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

4.1.4 Step 3: Upload file to Object Store

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

 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"

4.1.5 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:

 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
```

4.2 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.

4.2.1 Software Prerequisites

Python 3.10 (Note this is used to access data elements from the API JSON responses)

4.2.2 File Upload Using File Upload Utility

To upload files using the File Upload utility, follow these steps:

- 1. Extract the FILEUPLOAD_UTIL.zip file under the \$HOME directory.
- 2. Place the Data Loader input file in the **\$HOME/FILEUPLOAD_UTIL** directory.



Name	Date modified	Туре	Size	
S Env_setup.sh	2/3/2023 3:08 PM	Shell Script	1 KB	
input_20150101_filename_example_8004.txt	8/24/2022 9:45 PM	Text Document	1 KB	
PBSMCS_PAR_fileupload.sh	2/3/2023 3:08 PM	Shell Script	4 KB	
Readme.docx	2/3/2023 1:07 PM	Microsoft Word Doc	318 KB	

Figure 4-1 Data Loader Input File Path

 Edit the Env_setup.sh file to update the ENV variables. For more information to get the IDCS URL, see Get the OAuth Client ID and Client Secret for getting the URL.

IDCS_URL=<idcs-url>/oauth2/v1/token

Example: https://

idcs-0cb0c2b3ba624afca67467fd5eb9db49.identity.xxyy.xxyyy.com/oauth2/v1/ token

ENCODED=<OAuth Client ID>:<OAuth Client Secret> (Encode them using **base64encode**)

See Get the OAuth Client ID and Client Secret for getting the URL. for the Client ID and Client Secret details.

Example:

ZnRwcWExMDEyMzEtcHJkX0FQUEIEOjBkMmU5MDBiLTIhYjItNGFmOS050WM 0LTEwNTYyMDVkYWYwNQ==

USERNAME = < App login user>

Note: <App login user> should have FILE_UPLOAD role mapped.

PASSWORD=<App login password>

TENANT=<abcdef-prd>

PBSMHOST=<PBSMhostdetail>

Example : dc.pbsmcloud.us-xxxxx -1.ocs.oc-test.com

Figure 4-2 PBSM Host Details

```
EDIT - Start

IDCS_URL=<IDCS_URL>/oauth2/v1/token

ENCODED=ZnRwcWExMDEyMzEtcHJkX0FQUE1E0jBkMmU5MDBiLT1hYjItNGFmOS050WM0LTEwNTYyMDVkYWYwNQ==

USERNAME=xxxxxxx

PASSWORD=yyyyyyy

TENANT=abcdef-prd

PBSMH0ST=PBSMhostdetail

EDIT - END
```

4. Execute PBSMCS_PAR_fileupload.sh with parameters -

filename<space>filesize<space>mimetype. File size must be in bytes. 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, execute the script for each file.

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



-																			
÷	PBSMCS_	PAR_T1	eupload	sh ing	ut_20150101,	filename_e	xamp1e_80	07.txt]	34 text	/plain									
File	Name -	input_	0150101	filen	me_example_	1007.txt													
rile	size -	334																	
1005	URI -	text/p	ain							1/03	ath2/w1/	/token							
Enco	ded sec	ret - 1	DRACHEN	MOEYMER	CCH3kXOPQUE	E018kMmUSM	DB1LT1hY1	ItNGFm05	OSOWHOL	TENNTYMOW	k YW YMNQ#	E GH, etc.							
User	name -	ftp_qaa	ser																
E31.9	Upload	URL -	https://	<u> </u>						/01111	s-servic	00/v1/fi	le/uploa	dfile/p	FURL				
Tena	AE ID -		cpas//	-						autorities a	service	144/1110	(/acan						
Gen	erating	Accest	Token	for fil	le input_201	50101_filen	ane, exang	1e_8007.	txt										
Gen OtFw GMyY JQ51 2F02 m1kb Jd3N jpb12 Dkw0 k1w2 S522 bkw0 k1w2 S522 bkw0 k1w2 S522 bkw0 k52 k52 k52 k52 k52 k52 k52 k52 k52 k52	erated SISIMFS SISIMFS JNITYY TPOLES INVINE KFIGHT WINT WINT WINT WINT WINT SIGIRFU WINT WINT WINT WINT WINT WINT WINT WINT	Access 2y1611: NGFW721 2w50x22 bjpvCGP bWVk7X8 LC3jbG bhRp721 MT1jMT2 0TJk7T1 JW71dWT JWRNdW bhQ1LC2 FM0E2s1 7c_ARL	Token i DTMJUZIM L2NZQN2 Ikijoi2n MGawRtok MGawRtok NWRtaw IbnRfdG YDRQLC ZMJQ3ZD IbnRfdG YDRQLC ZMJQ3ZD IbnRfdG YDRQLC CASH IbnRfdG YDRQLC SASH IbnRfdG YDRC SASH IbnRfdG YDRC SASH IbnRfdC YDRC SASH IbnRfdC YDRC SASH IbnRfdC YDRC SASH IbnRfdC YDRC SASH IbnRfdC YDRC SASH IbnRfdC YDRC SASH IbnRfdC YDRC SASH IbnRfdC YDRC SASH IbnRfdC YDRC SASH IbnRfdC YDRC SASH IbnRfdC YDRC SASH IbnRfdC YDRC SASH IbnRfdC YDRC IbnRfdC YDR	s ey34% i0.ey316 i0.ey316 i0.ey316 i0.e251	xxjuzIN164 2YyX3861g61 KX106HLC3 MENDELC43 X126HLV2xp MENY2pp2006 MEY25261 KX106HV2xp MENY2pp2006 MEY25261 KX10130 KX104 XX02 XX02 KX104 XX02 XX02 XX02 XX02 XX02 XX02 XX02 XX	Cldia2SrQUR Dwllemjirtw Swi013odHR KOPQUETEI1w KOPQUETEI1w KOPQUETEI1w CSS2ZNA2AV SIMEWGNIMGM CIWI3MR ThenR SISSEITES SISEITES SISEITES SISEITES NOBMEZEMJQ SISSES SEF45D1R21g VSQSespILft	\$TUZIMIH1 vQ2bgv2Pm wC2pgCL1wV idKNIC19p vCGM6sikRt aIiwidXNI yV5NiHTYy pdHkgR09t DUyBm0HEx 2CGXheW5h DUyBm0HEx 2CGXheW5h D0H8z01wv 2N22EhWV 2N22EhWV 2N22EhWV 2N22EhWV 3N22EHWV 3N22EH	QipkeTY1 byIsInN1 awR1bnRp cOPkbw1u OnQudXN1 c19Yy21k NGPm12E2 NY1uIEFk NTEWTI2 bwU10130 NZEX2012 NZEX20 NZEX2012 NZEX20 NZEX20 NZEX20 NZEX20 NZEX20 NZEX20 NZEX20 NZEX20 NZEX20 NZEX20 NZEX20 NZEX20 NZEX20 NZEX20 NZEX20 NZEX NZEX20 NZEX20 NZEX NZEX NZEX NZEX NZEX NZEX NZEX NZEX	c3o4vzd v1162m2 dHku833 (2jpOcnv c15hdXR 1jo162N Wr20222 BWTu2XN WFAg1FF NDIx2WE 5.TB1v3B XSw1dXN c2VWF21 QT2Y0Bg Iz9QyPPS	EVwhKLOS 441 Ob F9 krXV220 TLC3 zdw3 fdk ozbis Olawhidd p ZDE udwhi e kraw fowr fno Ober F0b 31 133 (LC30 ZdShbr BVVN FU1 2 s1 41 kr V1 YVX12 1 c 19 p 2 c1 61 Wr 2 1 N2 51 1 c 19 p 2 c1 61 Wr 2 1 N2 51 9 d5 P9_ZX 941	ldvv0had xIILC3Lc wQuY29tX Hlv25I6I SUgdX3u0 ISy2Wdpb OkiLC3y2 Swi2XhwI NQI01322 Swi2XhwI NQI01322 Swi2XhwI NQI01322 Swi2XhwI NQI01322 Swi2XhwI NQI01322 Swi2XhwI Si20w25Y Si20w25Y Si20w25Y Si20w25Y	dx 4w4k t c2vyx2xv xc81c31 Invz2x1 Invz2x1 Invz2x1 Sofw2zPs 24xi, 15h Didpb35 f JoxNjg0 Cox2LT8 J dH3128v1 dH8z01wv x2niy2x1 dH8z01wv x2niy2x1 TY3Ncy3 bcg0b1p1 SngXNaRp	LC34N0Qi Y2F8Z5I6 C2VyX3R1 LC32Y29w 2006dC5z Y3F5Y25I6 NTc200k2 Y3F3SMIZ C3V1X21h XC9p26Nz Z0V1Y31k Z0Q2YNVh Z2Q2YNVh Z2Q2YNVh Z2Q2YNVh XC9p26Nz Z2Q1ZNI5 Y2Y14Eqw _18xDE40	DiJRetwi ImvuIiw DefruddSi ZSIGInVy ZSIGINVy ZSROawSY Yw42aXpr Inv2LwF LC3pYXQ Yw62aXpr ZNC2LwF V3G5Lm Ii ZjkynThi Zjkyn	icijotogi cilvinoj dojovcov cilvinoj dojovcov cojszavu dojezne cojszavu dojezne dojezne cojszavu dojezne doj	skQXpxYv viojq4mc lp2GuRto vGauRto vGauRto tGdz2ydE	h2Nwx0cv widxvlci 8jyj8j4W dgudxvlcn M5awRton dguGhb2 v2zx3fbG v5zmAsaw v1vj3kvji Shbwu1LC shbwu1LC shbwu1LC x2M15vvz wi12xp2W R1dj1ub2 hpwrv11T 2p50bu7A	vnQjcwSE SozwShbnn IzyHE2Mjj WfciBlom Qu2G1h22 h3N2V2M2 Fu2yIGInf VudP3ndw QSIIwian Jwcm1tV0 dm2DV1Vj 1xyWR1d1 SOQXBWUM MScwFF2N MScwFF2N SO227650	UiLC3rawQii QubeFt2526 Ah2whh3c0 46b38j0n1ki 5vc3Rp/3wfi Nu2iwidXN1 1ktjoiM3A2 Nu2iwidXN1 1ktjoiM3A2 5jb20iLC32 Ph20iM32 9b20iLC32 Ph20p10 2012 Ph20p10 2012 Ph20p10 2012 Ph20p10 2012 Ph20p10 2012 Ph20p10 2012 Ph20p10 2012 Ph20p10 2012 Ph20p10 2012 Ph20p10 2012 Ph20p10 2012 Ph20p10 2012 Ph20p10 2012 Ph20p10 2012 Ph20p10 2012 Ph20p10 2012 Ph20p10 2012 Ph20p10 2012 Ph20p10 2012 Ph20p10 Ph20P10 Ph20
PAR	URL Ge	neratio	n is su	ccessfu	1 for input.	20150101_f	ilename_e	xample_8	007.txt										
								Condition	a éternit a		PRO INCOM	a Section	and the first	Contraction data		1 64 A A	a set Tarakas	dal same a	. Marshur ala
ftpo	A101231	-prd_d	famore	TO CHEPT			A100000	=bde-835	1-25291	1a07002	PARAKREA	er. de l'un	PATON2NW	Jeoxoza:	gzeuogt	coet/n/c	raciegou	oevcorp/	e/msgou_por
FIL	E ID is	7985																	
	uploa	ding th	e File	to obje	ct store **														
26	Total	% Red	terived 3	xferd	Average Spi	ed Time	Time	Time	Curren										
100	334		0 100		Dlead upl	ad Total	Spent	Left	Speed										
100	334		0 200	334		12 0100102	0100106		101										
	**Scann	ing the	File "																
24	lotal	The Red	retived h	Aferd	Dload Upl	and Total	Chant.	1100	Speed	ie.									
100	34		34 0		0	0::	0:02:14		- 9	["payload":	:"Scanne	ed succe	ssfully"						
***	File Up	load is	Succes	sful pl	ease check	File Upload	/ Downlo	ad ur+++											

5. After the execution is completed, check the File Upload/Download UI to confirm that the uploaded file is appearing.

Figure 4-4 File Upload/Download UI

File Up	load/Do	wnload				
4	Drag and I Select a file or	Drop drop one here +	Prefix Selected File	6		
			Upload	Get PAR URL		
8004		0				
File 10	Prefix	File Name	Stripe	Name	Uploaded Date	0
7981		input_20150101_filename_example_8004.bt	defaul	ı	03-FEB-23 09:38:51 AM	



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.

5.1 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

5.1.1 End Point Details

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

5.1.2 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
eyJ4NXQjUz11Ni16Ildia25rQUR5TUZIMlhlQ1pKcTY1c3o4VzdEVWhKa0s4MldYY0ha
dk4wWkkiLCJ4
...
sQXj0iohsSIEmQXVwwjhhqnc4eJNnmCjx8Tb7TXjx1MIQLeOIcfrIj5gkzoMKX94_7US
xHv-6LhBzw'
```

5.1.2.1 Request JSON Parameters

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

Name	Туре	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
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 less than 7 GB.

Table 5-1 Request JSON Parameters



Name	Туре	Required	Description
mimeType	STRING	Yes	The mime type to be uploaded.
			The following mime types are allowed:
			 Text/CSV
			 Text/plain
			• DAT

Table 5-1 (Cont.) Request JSON Parameters

Request JSON Sample

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

5.1.2.2 Response JSON Parameters

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

Table 5-2 POST JSON Response

Name	Туре	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-hzigiAoUU01yEKnuk0YGs05L172gt_woZAgqNFYmUFQeexV3BDfT097mhBI/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"
    }
}
```



5.1.3 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

5.2 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.

5.2.1 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 '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 '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"}}
```

5.2.2 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_op0Zk41vKGDxxeXGhLq7/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"}}
```

