Application Programming Guide

Version 9.4.0

October 30, 2024 | 3010240905|

Copyrights and Trademark Notices

Copyright © 2024 Telestream, LLC and its Affiliates. All rights reserved. No part of this publication may be reproduced, transmitted, transcribed, altered, or translated into any languages without written permission of Telestream, LLC. Information and specifications in this document are subject to change without notice and do not represent a commitment on the part of Telestream. Specifications subject to change without notice.

Telestream, Aurora, CaptionMaker, CaptureVU, Cerify, Content Manager, ContentCentral, Cricket, DIVA, DIVAdirector, DIVADocs, DIVAGrid, DIVANet, DIVAProtect, DIVASolutions, Episode, Encoding Intelligence, Episode, FLEXVU, Flip4Mac, FlipFactory, Flip Player, Geminus, Glim, GraphicsFactory, Inspector, IQ & Design, Kumulate, Lightspeed, MassStore, MassTech, MetaFlip, Post Producer, Prism, ScreenFlow, Sentry, Singulus, Split-and-Stitch, Stay Genlock, Surveyor, Tempo, TrafficManager, Vantage, Vantage Cloud Port, VOD Producer, and Wirecast are registered trademarks of Telestream, LLC and its Affiliates and its Affiliates.

Argus, ContentAgent, Cricket, e-Captioning, Inspector, iQ, iVMS, iVMS ASM, MacCaption, Pipeline, Switch, and Vidchecker are trademarks of Telestream, LLC and its Affiliates. All other trademarks are the property of their respective owners.

Adobe. Adobe® HTTP Dynamic Streaming Copyright © 2014 Adobe Systems. All rights reserved.

Apple. QuickTime, MacOS X, and Safari are trademarks of Apple, Inc. Bonjour, the Bonjour logo, and the Bonjour symbol are trademarks of Apple, Inc.

Avid. Portions of this product Copyright 2012 Avid Technology, Inc.

CoreOS. Developers of ETCD.

Dolby. Dolby and the double-D symbol are registered trademarks of Dolby Laboratories Licensing Corporation.

Fraunhofer IIS and Thomson Multimedia. MPEG Layer-3 audio coding technology licensed from Fraunhofer IIS and Thomson Multimedia.

intoPIX and Fraunhofer. Notice under 35 U.S.C. §287(a): This product or service includes JPEG XS compliant features that are covered by patents in the United States and in other jurisdictions owned by intoPIX SA (“intoPIX”) and/or Fraunhofer-Gesellschaft zur Foerderung der angewandten Forschung E.V. (“Fraunhofer”) and listed at HYPERLINK "http://www.jpegxspool.com"www.jpegxspool.com. Additional patents may be pending in United States and elsewhere.

Google. VP6 and VP8 Copyright Google Inc. 2014 All rights reserved.

MainConcept. MainConcept is a registered trademark of MainConcept LLC and MainConcept AG. Copyright 2004 MainConcept Multimedia Technologies.

Manzanita. Manzanita is a registered trademark of Manzanita Systems, Inc.

MCW. HEVC Decoding software licensed from MCW.

MediaInfo. Copyright © 2002-2013 MediaArea.net SARL. All rights reserved.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Microsoft. Microsoft, Windows NT|2000|XP|XP Professional|Server 2003|Server 2008 |Server 2012|Server 2016|Server 2019|Server 2022, Windows 7, Windows 8, Windows 10, Windows 11, Media Player, Media Encoder, .Net, Internet Explorer, SQL Server 2005|2008|2012|2016|2019, and Windows Media Technologies are trademarks of Microsoft Corporation.

NLOG, MIT, Apache, Google. NLog open source code used in this product under MIT License and Apache License is copyright © 2014-2016 by Google, Inc., © 2016 by Stabzs, © 2015 by Hiro, Sjoerd Tieleman, © 2016 by Denis Pushkarev, © 2015 by Dash Industry Forum. All rights reserved.

SharpSSH2. SharpSSH2 Copyright (c) 2008, Ryan Faircloth. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer:

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of Diversified Sales and Service, Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Swagger. Licensed from SmartBear.

Telerik. RadControls for ASP.NET AJAX copyright Telerik All rights reserved.

VoiceAge. This product is manufactured by Telestream under license from VoiceAge Corporation.

x264 LLC. The product is manufactured by Telestream under license from x264 LLC.

Xceed. The Software is Copyright ©1994-2012 Xceed Software Inc., all rights reserved.

ZLIB. Copyright (C) 1995-2013 Jean-loup Gailly and Mark Adler.

Other brands, product names, and company names are trademarks of their respective holders, and are used for identification purpose only.

MPEG Disclaimers

MPEGLA MPEG2 Patent

ANY USE OF THIS PRODUCT IN ANY MANNER OTHER THAN PERSONAL USE THAT COMPLIES WITH THE MPEG-2 STANDARD FOR ENCODING VIDEO INFORMATION FOR PACKAGED MEDIA IS EXPRESSLY PROHIBITED WITHOUT A LICENSE UNDER APPLICABLE PATENTS IN THE MPEG-2 PATENT PORTFOLIO, WHICH LICENSE IS AVAILABLE FROM MPEG LA, LLC, 4600 S. Ulster Street, Suite 400, Denver, Colorado 80237 U.S.A.

MPEGLA MPEG4 VISUAL

THIS PRODUCT IS LICENSED UNDER THE MPEG-4 VISUAL PATENT PORTFOLIO LICENSE FOR THE PERSONAL AND NON-COMMERCIAL USE OF A CONSUMER FOR (i) ENCODING VIDEO IN COMPLIANCE WITH THE MPEG-4 VISUAL STANDARD ("MPEG-4 VIDEO") AND/OR (ii) DECODING MPEG-4 VIDEO THAT WAS ENCODED BY A CONSUMER ENGAGED IN A PERSONAL AND NON-COMMERCIAL ACTIVITY AND/OR WAS OBTAINED FROM A VIDEO PROVIDER LICENSE IS GRANTED OR SHALL BE IMPLIED FOR ANY OTHER USE. ADDITIONAL INFORMATION INCLUDING THAT RELATING TO PROMOTIONAL, INTERNAL AND COMMERCIAL USES AND LICENSING MAY BE OBTAINED FROM MPEG LA, LLC. SEE HTTP://WWW.MPEGLA.COM.

MPEGLA AVC

THIS PRODUCT IS LICENSED UNDER THE AVC PATENT PORTFOLIO LICENSE FOR THE PERSONAL USE OF A CONSUMER OR OTHER USES IN WHICH IT DOES NOT RECEIVE REMUNERATION TO (i) ENCODE VIDEO IN COMPLIANCE WITH THE AVC STANDARD ("AVC VIDEO") AND/OR (ii) DECODE AVC VIDEO THAT WAS ENCODED BY A CONSUMER ENGAGED IN A PERSONAL ACTIVITY AND/OR WAS OBTAINED FROM A VIDEO PROVIDER LICENSED TO PROVIDE AVC VIDEO. NO LICENSE IS GRANTED OR SHALL BE IMPLIED FOR ANY OTHER USE. ADDITIONAL INFORMATION MAY BE OBTAINED FROM MPEG LA, L.L.C. SEE HTTP://WWW.MPEGLA.COM.

MPEG4 SYSTEMS

THIS PRODUCT IS LICENSED UNDER THE MPEG-4 SYSTEMS PATENT PORTFOLIO LICENSE FOR ENCODING IN COMPLIANCE WITH THE MPEG-4 SYSTEMS STANDARD, EXCEPT THAT AN ADDITIONAL LICENSE AND PAYMENT OF ROYALTIES ARE NECESSARY FOR ENCODING IN CONNECTION WITH (i) DATA STORED OR REPLICATED IN PHYSICAL MEDIA WHICH IS PAID FOR ON A TITLE BY TITLE BASIS AND/OR (ii) DATA WHICH IS PAID FOR ON A TITLE BY TITLE BASIS AND IS TRANSMITTED TO AN END USER FOR PERMANENT STORAGE AND/OR USE. SUCH ADDITIONAL LICENSE MAY BE OBTAINED FROM MPEG LA, LLC. SEE HTTP://WWW.MPEGLA.COM FOR ADDITIONAL DETAILS.

Limited Warranty and Disclaimers

Telestream, LLC (the Company) warrants to the original registered end user that the product will perform as stated below for a period of one (1) year from the date of shipment from factory:

Hardware and Media—The Product hardware components, if any, including equipment supplied but not manufactured by the Company but NOT including any third party equipment that has been substituted by the Distributor for such equipment (the "Hardware"), will be free from defects in materials and workmanship under normal operating conditions and use.

Warranty Remedies

Your sole remedies under this limited warranty are as follows:

Hardware and Media—The Company will either repair or replace (at its option) any defective Hardware component or part, or Software Media, with new or like new Hardware components or Software Media. Components may not be necessarily the same, but will be of equivalent operation and quality.

Software Updates

Except as may be provided in a separate agreement between Telestream and You, if any, Telestream is under no obligation to maintain or support the Software and Telestream has no obligation to furnish you with any further assistance, technical

support, documentation, software, update, upgrades, or information of any nature or kind.

Restrictions and Conditions of Limited Warranty

This Limited Warranty will be void and of no force and effect if (i) Product Hardware or Software Media, or any part thereof, is damaged due to abuse, misuse, alteration, neglect, or shipping, or as a result of service or modification by a party other than the Company, or (ii) Software is modified without the written consent of the Company.

Limitations of Warranties

THE EXPRESS WARRANTIES SET FORTH IN THIS AGREEMENT ARE IN LIEU OF ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. No oral or written information or advice given by the Company, its distributors, dealers or agents, shall increase the scope of this Limited Warranty or create any new warranties.

Geographical Limitation of Warranty—This limited warranty is valid only within the country in which the Product is purchased/licensed.

Limitations on Remedies—YOUR EXCLUSIVE REMEDIES, AND THE ENTIRE LIABILITY OF TELESTREAM, LLC WITH RESPECT TO THE PRODUCT, SHALL BE AS STATED IN THIS LIMITED WARRANTY. Your sole and exclusive remedy for any and all breaches of any Limited Warranty by the Company shall be the recovery of reasonable damages which, in the aggregate, shall not exceed the total amount of the combined license fee and purchase price paid by you for the Product.

Damages

TELESTREAM, LLC SHALL NOT BE LIABLE TO YOU FOR ANY DAMAGES, INCLUDING ANY LOST PROFITS, LOST SAVINGS, OR OTHER INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF YOUR USE OR INABILITY TO USE THE PRODUCT, OR THE BREACH OF ANY EXPRESS OR IMPLIED WARRANTY, EVEN IF THE COMPANY HAS BEEN ADVISED OF THE POSSIBILITY OF THOSE DAMAGES, OR ANY REMEDY PROVIDED FAILS OF ITS ESSENTIAL PURPOSE.

Further information regarding this limited warranty may be obtained by writing:
Telestream, LLC
848 Gold Flat Road
Nevada City, CA 95959 USA

You can call Telestream during U. S. business hours via telephone at (530) 470-1300.

Regulatory Compliance

Electromagnetic Emissions: EN 55032, IEC/EN 61000-3-2, IEC/EN 61000-3-3, FCC Part 15 Subpart B, ICES-003, VCCI 32-1, AS/NZS CISPR 32

Electromagnetic Immunity: EN 550535

Safety: IEC 62368-1, EN 62368-1, CSA C22.2 No. 62368-1-14, UL 62368-1

California Best Management Practices Regulations for Perchlorate Materials:
This Perchlorate warning applies only to products containing CR (Manganese Dioxide) Lithium coin cells. Perchlorate Material-special handling may apply. See www.dtsc.ca.gov/hazardouswaste/perchlorate.

Obtaining Support | Information | Assistance

Contact Telestream for support, information or assistance, as indicated.

Resource	Contact Information
DIVA Technical Support | Information | Assistance | FAQ | Forums | Upgrades	Web site: telestream.net/telestream-support/DIVA/support.htm Web site: telestream.net/telestream-support/DIVA/support.htm Support Email. support@telestream.net US Phone: 877-257-6245 International Phone: +1 530-470-2036 Support hours: Monday - Friday, 7am - 6pm Pacific time. P1 Support: 24 x 7. Terms and times of support services vary, per the terms of your current service contract with Telestream. Depending on problem severity, Telestream responds within 24 business hours. For P1 issues, Telestream responds within 1 hour. Please see the Maintenance & Support Guide for definitions.
Product Licensing	Web site: telestream.net/telestream-support/DIVA/support.htm License Email: license@telestream.net
Telestream, generally	Web site: telestream.net Sales and Marketing Email: info@telestream.net

Resource	Contact Information
Telestream Reseller Support	If you purchased your product from a reseller, please contact your reseller for support.
International Reseller Support	Website: telestream.net See the website for your regional authorized Telestream reseller.
Telestream Technical Writers	Email: techwriter@telestream.net If you have comments or suggestions about improving this document or other Telestream documents—or if you've discovered an error or omission, please email us.

DIVA Documentation Library

For more information about DIVA, browse the DIVA Documentation Library at telestream.net/telestream-support/DIVA/support.htm.

Contents

Introduction 11

DIVA Installation Instructions 11
DIVA Jobs (Requests) 11
- Archive Job Terms 12
- Restore Job Terms 13
- Partial Restore Job Terms 13
- Delete Object Job Terms 13

Operational Boundaries 14
- Supported API 14
- Number of DIVA Connections 14
- Number of Simultaneous DIVA Jobs 14
- Number of API Tasks 14
- Special Authorized Characters 14
- Maximum Number of Allowed Characters 15
- File Path Limitations 15

DIVA REST API 16

Main DIVA REST API Calls 16
- DIVA REST APIs on Swagger 17
- Start Configuring the DIVA REST API 17
- Sample Python Program Using DIVA REST API 19
- DIVA REST API Secure Communications 20

DIVA Data Service API 21
- DIVA Manager Endpoints 22

DIVA Connect REST API 22
DIVA REST API Workflows 23
- Authentication Token Workflow 23
- Roles 24
- DIVA REST API Workflow Guidelines 25
- DIVA Job Status Codes 26
- Partial Restore Job Formats and Manager Responses 26

Python DIVAScript 30

Python DIVAScript 30
DIVAScript Commands Supported by DIVA REST API 30
DIVAScript Configuration File Settings 31
Running DIVAScript in Server Mode 31
DIVAScript Command-Line Mode Requirements 31
DivaScript Command Syntax 32

Introduction

This guide provides information and instruction for software developers who create applications using DIVA. This guide contains information about configuring and operating DIVA for various programming architectures. Consult it when installing, configuring, or operating DIVA for any of these architectures.

Topics

- DIVA Installation Instructions
- DIVA Jobs (Requests)
- Operational Boundaries

DIVA Installation Instructions

For instructions on installing DIVA, see the *DIVA Installation and Administration Guide* on the Telestream website, here: https://www.telestream.net/telestream-support/diva/support.htm.

DIVA Jobs (Requests)

Archive, Restore and Delete are the basic DIVA jobs. The core concept of DIVA is data preservation.

Note: In DIVA programming, we now use the term *job* instead of *request*. *Job* also is the official term used in programming for the DIVA Web App and all DIVA documentation. However, documentation for earlier versions of DIVA, which use *request*, still uses the term *request*.

Note:

Note: Telestream has deprecated the DIVA C++ API and the DIVA Java API. Use DIVA REST API. DIVA REST API supports improved DIVA functionality and the latest features. For documentation of the deprecated APIs, see the *DIVA Application Programming Guide*, versions 9.3 and earlier.

Note:

Archive Job Terms

Object

DIVA stores objects. An object is a set of files referring to an asset or a clip. An object can be made of one file, typically an MXF file, or with several files like reference MOV format (one video file, several audio files), or DPX format.

Identified an object by a Name and a Collection (category). Choose names for Object Name and Collection as desired. DIVA checks only that the Object Name + Collection combination is unique.

Collection

In DIVA, a Collection is like a name extension, not to be confused with a Tape Group. Any name can be used for a Collection. Telestream recommends using the application or company name so we can identify who has sent a job. Should the same Object Name be used for different clips (typically hi-res and low-res), use a different Collection to distinguish those clips.

Files

The Files parameter provides the names of the files of the object to be archived; each name can contain a relative path to the file location.

Media Name

Media Name is the DIVA device used for storing the object. The device can be a disk, a tape or cloud storage. Each of these devices can have multiple names based on partitioning (for example, DIVAGRID, NAS-STORAGE, TAPE_SPORTS_MAIN, TAPE_SPORTS_BKP, CLOUD_PROGRAM, CLOUD_PROMOS, and so on).

You can retrieve the list of all Arrays and Tape Groups can from DIVA, without necessarily exposing these to the end user. Check with the customer and the DIVA Project Manager about which Media to expose to the end user.

You can also use a Storage Plan as a storage device. For details, see the *DIVA Storage Policy Manager* guide.

Unmanaged Storage Repository Name

The Unmanaged Storage Repository Name is the content server name where DIVA will archive from. It must be the same name as in the DIVA configuration. Confirm this with the customer or DIVA Project Manager for this list.

Source Path Root

The Source Path Root is the File Path Root where the content objects are located. By default, DIVA uses the default File Path Root configured for that source in the DIVA configuration.

Note: You can obtain the source list using the `GET /servers` API call.

Quality of Service

You can leave the Quality of Service parameter at the default setting.

Priority

The Priority (between 1 and 100 highest) can either remain at the default, or a value can be specified.

Delete From Source

If the Delete From Source option check box is selected, then that parameter will delete the asset just archived from the Source Server, but only if the archive was successful.

Restore Job Terms

You must specify the following terms for a Restore Job:

- Object Name
- Object Collection
- Unmanaged Storage Repository Server Name
- The File Path Root; if empty, DIVA will take the File Path Root used during the Archive job and will overwrite the object if it already exists, unless the Do Not Overwrite option is specified.

Partial Restore Job Terms

The Partial Restore terms are the same as the Restore terms with the following additional options:

- Offset or Timecodes (In/Out) or File List
- Partial Restore will create a new clip name because it generates a new clip created with a portion of the original clip.

Delete Object Job Terms

A Delete Object Job deletes all copies of that object whether they are on disk, tape (in the tape library or external), or in the cloud. The Object Name and Object Collection must be specified.

Note: Deleting an object implies it cannot be recovered from the storage media after deletion.

Operational Boundaries

Supported API

Telestream has deprecated the DIVA C++ API and the DIVA Java API. Use DIVA REST API. DIVA REST API supports improved DIVA functionality and the latest features.
For documentation of the deprecated APIs, see the *DIVA Application Programming Guide*, versions 9.3 and earlier.

Number of DIVA Connections

The number of connections to DIVA is limited by DIVA and set in the DIVA configuration file. The default configuration limit is 200. This includes connections to GUIs, Actors and all API clients. When the configured limit is reached, the API doesn't create additional connections.

See the diva.conf and manager.conf configuration files for more information.

Number of Simultaneous DIVA Jobs

The maximum number of simultaneous jobs processed by DIVA is configurable in the diva.conf file as the value of the `DIVA_MAX_SIMULTANEOUS_REQUESTS` parameter. The default value has been raised from two hundred to five hundred. The maximum number has been verified up to two thousand. Additional simultaneous jobs beyond the value set in this parameter are rejected by DIVA.

Number of API Tasks

The number of API tasks that will be accepted to the API Processing Queue is configurable in the diva.conf file as the value of the `DIVA_API_TASK_QUEUE_SIZE` parameter. The default value is two thousand and DIVA has been verified at this value. If the queue is full, subsequent commands are rejected.

Special Authorized Characters

Many jobs require alpha-numeric text terms. You can use special characters in these fields. If you use an invalid special character, the job is rejected. See the topic, *List of Authorized Special Characters* on this Swagger page: Manager API.

In a Windows. environment, file and folder names can't consist of one or more spaces, and can't contain a double-quotation mark.

Maximum Number of Allowed Characters

There is a maximum number of characters you can use for each job term. If these limits are exceeded, the job is rejected. See the table of maximum number of characters on this Swagger API page: Manager.

File Path Limitations

DIVA supports absolute path names on both Windows up to a maximum of 4000 characters. Relative path names are limited to 32676 characters on Windows systems (only).

A DIVA Windows local path is structured in the following order and terminated with a NUL character:

```
Drive_Letter:\Component_Name\Component_Name\File_Name.Extension
```

Here are example paths used in DIVA in Windows. The <NUL> character used in the example represents the invisible terminating null character for the current system code page and need not be entered. (The < and > characters are used to identify the NUL character by name, and must not be part of a path string.

Generic Path:

```
D:\Some_32676-Character_Path_String<NUL>
```

Actor Executable:

```
D:\diva\diva\Program\Actor\bin\diva.exe
```

DIVA Configuration File:

```
D:\diva\10\Program\conf\diva\diva.conf
```

DIVA REST API

This chapter provides details about programming the DIVA REST API.

Topics

- Main DIVA REST API Calls
- DIVA Data Service API
- DIVA Connect REST API
- DIVA REST API Workflows

Main DIVA REST API Calls

The following are the main DIVA REST API calls available. These are the minimum required to implement the basic DIVA REST API workflows.

- `POST /users/logins`
- `POST /users/logout`
- `GET /groups`
- `GET /arrays`
- `GET /object/info`
- `GET /objects/list`
- `GET /requests`
- `POST /requests/archive`
- `POST /requests/cancel`
- `POST /requests/delete`
- `POST /requests/partialRestore`
- `POST /requests/restore`
- `GET /requests/{requestId}`
- `GET /versions`

The Swagger definitions for these endpoints are located here:

https://127.0.0.1:8765/webjars/swagger-ui/index.html?urls.primaryName=data

https://127.0.0.1:8765/webjars/swagger-ui/index.html?urls.primaryName=manager

DIVA REST APIs on Swagger

The REST API Swagger page is home to all DIVA REST APIs. You can find other APIs using the Definition drop-down menu on the Swagger page.

The Swagger documentation for the REST API services is accessible using the following URL:
https://127.0.0.1:8765/webjars/swagger-ui/index.html

Important: Be sure to check the details in the Swagger API comments.

For the purposes of this document we will focus of the data and manager service required to login and submit jobs, respectively.

Start Configuring the DIVA REST API

To start configuring the DIVA REST API, do the following:

1. Obtain all automations and DIVA REST API calls that use these credentials from the person who created the user.

   During installation, a user is created by either the DIVA Installer, or manually by an administrator.

2. Call the `POST users/login` endpoint.

   DIVA opens the *`POST users/login`* dialog.

3. In the *userLogin* field, specify the login and password.

   For example:

   ```
   "emailAddress": "a@abc.com",
   "password": "password",
   ```

4. Click the *Execute* button.

   DIVA sends you a token.

5. In the *Responses* window, copy the contents of the bearer token (everything in quotes after `"Token":`)

6. To create a user, enter the token in the *Authorization* field.

7. In the *userReg* field, enter the user name, password, and role of the user.

   For example:

   ```
   "emailAddress": "a@abc.com",
   "password": "password",
   "role": "operator",
   ```

8. (Optional) Set a timeout value between 0 and 10080 minutes. Default: 1440 minutes.

   For example:

   `"sessionTimeout": 99999999`

   The session timeout is the number of minutes the user remains logged into DIVA.

Note: Call `GET /roles` to obtain a list of possible roles.

- All DIVA `GET` jobs require at least the `user` role.
- Archive, Restore (including N-Restore and Partial Restore) and Copy jobs require at least the `operator` role.
- Change Priority, Transfer, Eject, Insert, Export and Import jobs require at least the `advoperator` role.
- All other jobs require the *administrator* role.

You can now use the DIVA REST API to retrieve information from DIVA. Switch to the Manager endpoints to start using the DIVA REST API.

Retrieving All Configured Actors

To retrieve all configured Actors, do the following:

1. To get a list of all actors in the configuration, call `GET/actors`.

   DIVA opens the `GET/actors` dialog.

2. Click the *Execute* button.

   DIVA displays a list of actor settings in a *Responses* dialog.

Submitting a Job

To submit a job (for example, an Archive job, do the following:

1. Submit `POST /requests/archive`.

   DIVA opens the `POST /requests/archive` dialog.

2. Enter an authorization key into the `Authorization string` field with the bearer token as the value, as depicted in this job:

```
curl -X POST --header 'Content-Type: application/json' --header
'Accept: application/json' --header 'Authorization: Bearer
eyJhbGciOiJIUzUx-
MiJ9.eyJhdWQiOiI1MjM5YTcxOS1iYjAwLTQ5MWQtOGYxZi01ZjcxM2YxZWZiMj
MiLCJleHAiOjE2MjEzNTY0MDcsImlhdCI6MTYyMTI3MDAwNywiYXV0aG9yaXRpZ
XMiOlsic3lzYWRtaW4iXSwidXNlcm5hbWUiOiJzeXNhZG1pbiJ9.zZiKvEe-
```

```
3JjuOsJ-CDpW_32JKRefy54-wGwra_LABmUeuIhpWGEpHnT-Se5PXTFxv-
jDf2g9mgezKQIvIJzObzQ' -d '{ \
   "collectionName": "a", \
   "comments": "this is object a2", \
   "components": [ \
     "1.txt" \
   ], \
   "filePathRoot": "", \
   "media": "default", \
   "objectName": "a2", \
   "options": "", \
   "priority": 50, \
   "qos": 2, \
   "sourceServer": "wfm_ftp_sd_for_diva_test" \
 }' 'http://xxx.xx.xx.xx:xxxx/manager/requests/archive'
```

3. Go to the Swagger page for the job and click on the Example Value.

   Swagger opens the `POST/requests/archive` dialog, displaying the fields that must be specified for any job.
4. Specify the values for the *Authorization string* and *body* fields.
5. To submit the job, click *Execute*.

   DIVA opens the `POST /requests/archive` dialog.
6. To view a description of each field and a list possible values, click *Model* (next to the *Example Value* tab).

   DIVA opens the *Model* dialog.

For example, DIVA displays the list of possible QOS values and their meaning. A value of 2 signifies a QOS value of Direct-only.

Sample Python Program Using DIVA REST API

Here is a sample program to obtain all Actors from DIVA using Python:

```
import requests

url = https://127.0.0.1:8765/dataservice/users/login

headers = {
        "Content-Type": "application/json; utf-8",
        "Accept": "application/json"
}

json = {
  "username": "enter_the_username_here",
  "password": "enter_the_password_here"
}
```

```
response = requests.post(url, headers=headers, json=json,
verify=False)

token = response.json()["token"]

print(token)

url = https://127.0.0.1:8765/manager/actors?page=1&size=5

headers = {
        "Accept": "application/json",
        "Authorization": token
}

response = requests.get(url, headers=headers, verify=False)

print(response.json())
```

DIVA REST API Secure Communications

All DIVA REST API services communicate securely via HTTPS using certificates from the `DIVA\program\security` folder.

The DIVA security tool (`DIVA\Program\security\bin\DIVASecurityTool.bat`) generates certificates signed by the DIVA certificate authority (`DIVA_CA`).

The second security tool option automatically adds the IP address and the hostname of the server to the `alt_names` section at the end of the `DIVA\Program\DIVA_CA\DIVA_CA.cnf` file. Then, the security tool generates new certificates allowing IP addresses in the `alt_names` section.

After generating new certificates, you must restart all the DIVA services.

Clients using the REST APIs can communicate with the secure API Gateway on port 8765.

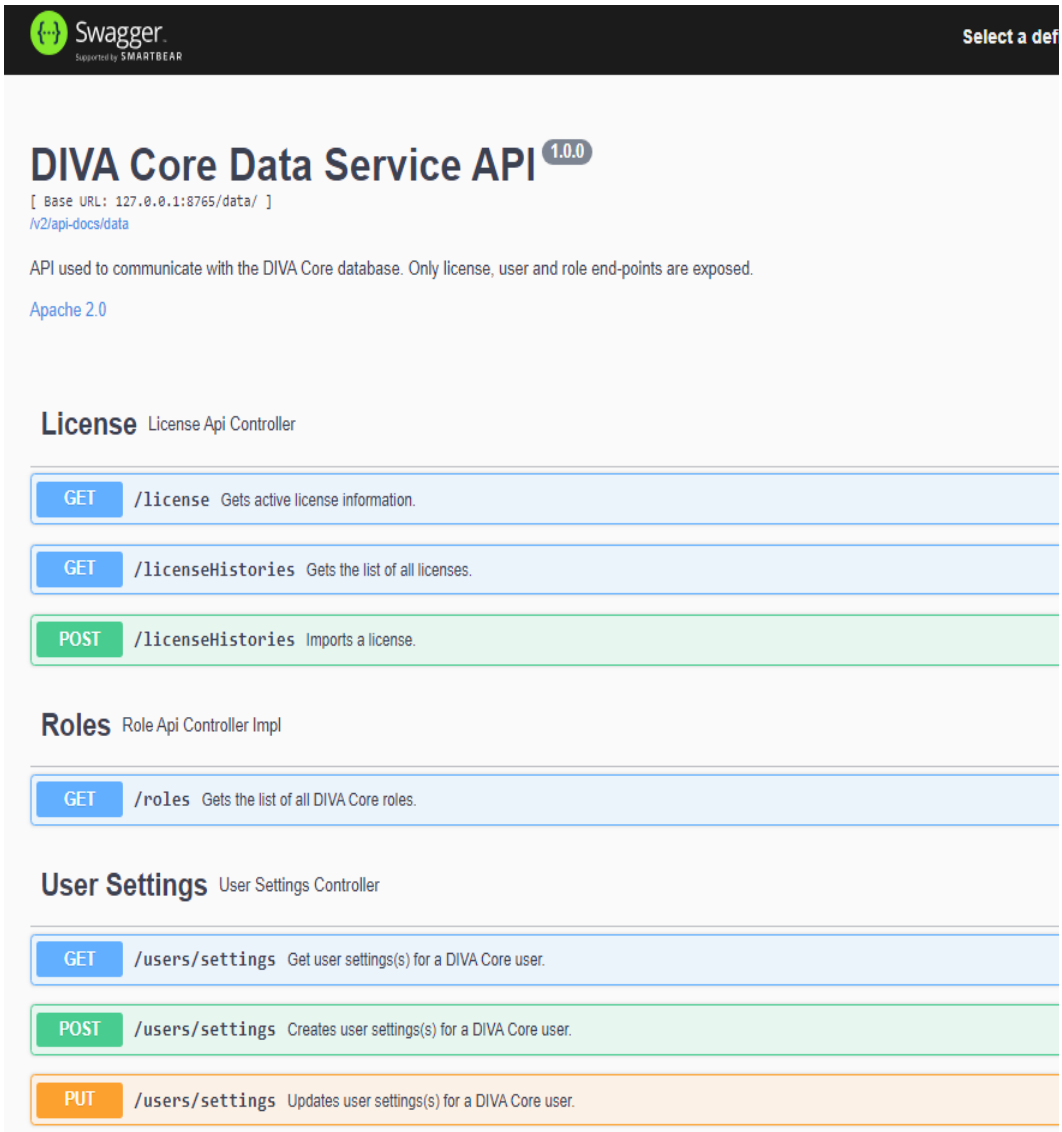
To see all services available on the Gateway, browse to `https://IP_Address:8765/swagger-ui.html`. To securely access the Swagger page and all child pages, the IP address must be in the `alt_names` section of `DIVA\Program\DIVA_CA\DIVA_CA.cnf`.

For assistance, contact Telestream Technical Support.

DIVA Data Service API

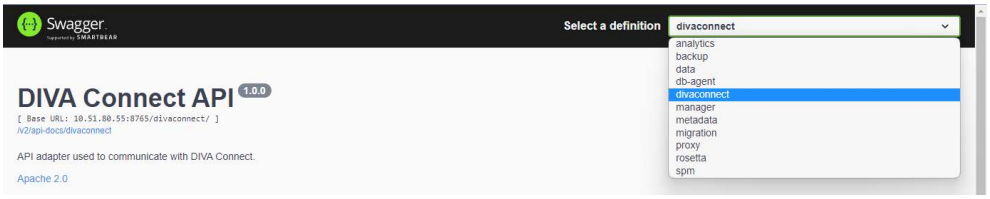
The DATA Service API communicates with the database. Only user, profile, and endpoints are exposed.

Use the DIVA Core Data Service Swagger page to manage users, roles and profiles. After a user is created through `POST /users`, that user can obtain an access token through `POST /users/login` that will be needed for all future communication; including accessing all DIVA resources available in the Manager Endpoints.

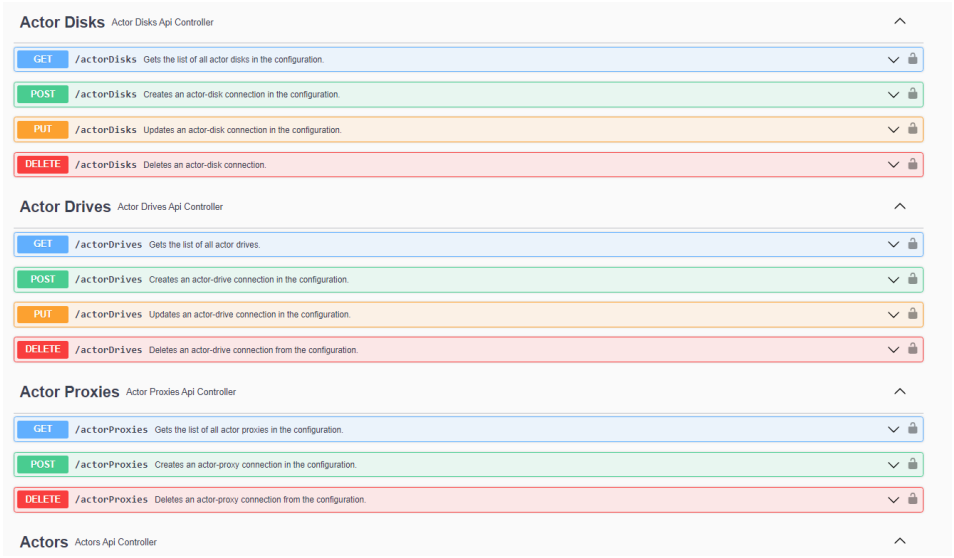


DIVA Manager Endpoints

The Data Service endpoints can be switched to other endpoints using the menu at the top of the page:



You use the API to communicate with DIVA. Use these endpoints to submit jobs and obtain information on DIVA resources and jobs.



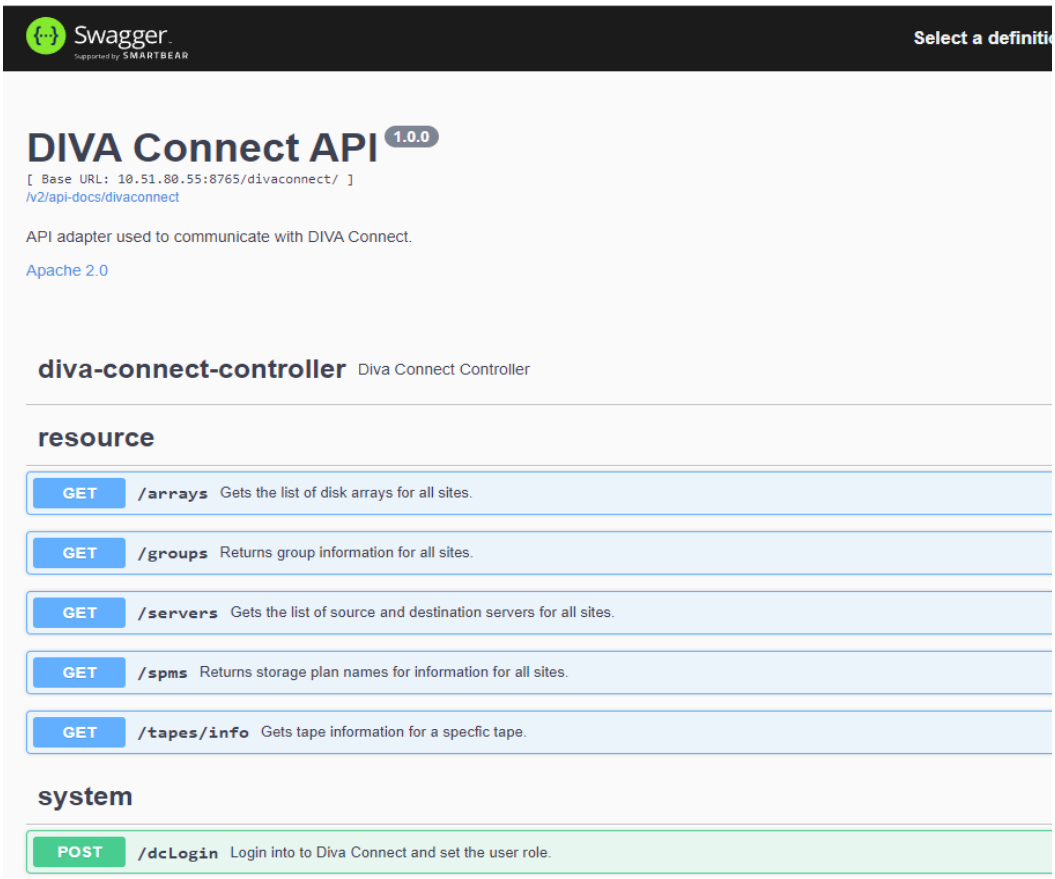
DIVA Connect REST API

You can use the DIVA REST API to send jobs and commands to DIVA Connect. The same port used for the Client Web Connections is used for this API. The DIVA REST API supports only MultiDIVA Mode.

During installation, a user is created either by the DIVA Installer, or manually by an administrator. Obtain all automations and API calls that use these credentials, from the person who created the user. Go to the POST users/login.

For detailed information, see the DIVA Connect documentation on the DIVA Support Portal at https://www.telestream.net/telestream-support/diva/support.htm

Swagger contains DIVA Connect REST API endpoints as shown here:



DIVA REST API Workflows

This chapter describes the DIVA REST API and Authentication Token Workflows. The DIVA REST API uses JWT (JSON Web Token) authentication specified in the authorization header of all jobs. To obtain the token, `POST to /users/login` on the data service; passing in the user name and password. There is a specific endpoint to get a authentication token and all the functions of the DIVA REST API require this token to function properly.

Authentication Token Workflow

The authentication phase is mandatory in order to get a token that will be used for any following API call. A token is configurable and valid for 1440 minutes (24 hours) by default. The maximum is 10080 minutes (7 days). It is advised to authenticate one time at the start of your application before the first call to a DIVA REST API, and then use that token as long as it is valid. Any HTTP job using an invalid or expired token will fail with HTTP error code 403 (access denied).

The following process is the authentication workflow.

1. Upon login the user will receive an authentication token.
2. An access token must be used to access secured endpoints. It will automatically expire after one day. Alternatively, a user may delete an access token by calling `/users/logout`.
3. When an access token expires or is deleted, the client is considered as logged out and must login again.

Roles

A user may belong to one of five roles; sysadmin, admin, advoperator, operator, or user.

A user may perform all basic GET operations including the following:

- `POST /users/login`
- `POST /users/logout`
- `PUT /users/{userName}/password`
- `GET /profile`
- `PUT /profile`
- `GET /users`
- `GET /roles`
- `GET ANY RESOURCE (for example, GET /actors)`

An Operator may perform all the operations of a user and the following additional operations:

- `POST /requests/archive`
- `POST /requests/restore`
- `POST /requests/copy`

An Advanced Operator (`advoperator`) may perform all the operations of an operator and the following additional operations:

- `PUT /requests`
- `POST /requests/transferFiles`
- `POST /requests/insertTape`
- `POST /requests/ejectTape`
- `POST /requests/repackTape`
- `POST /requests/exportTape`
- `POST /requests/importTape`

An Administrator (admin) may perform all operations of an `advoperator` and the following additional operations:

- `POST /requests/delete`
- `POST /requests/serverDelete`

A System Administrator (sysadmin) may perform all operations of an administrator and the following additional operations:

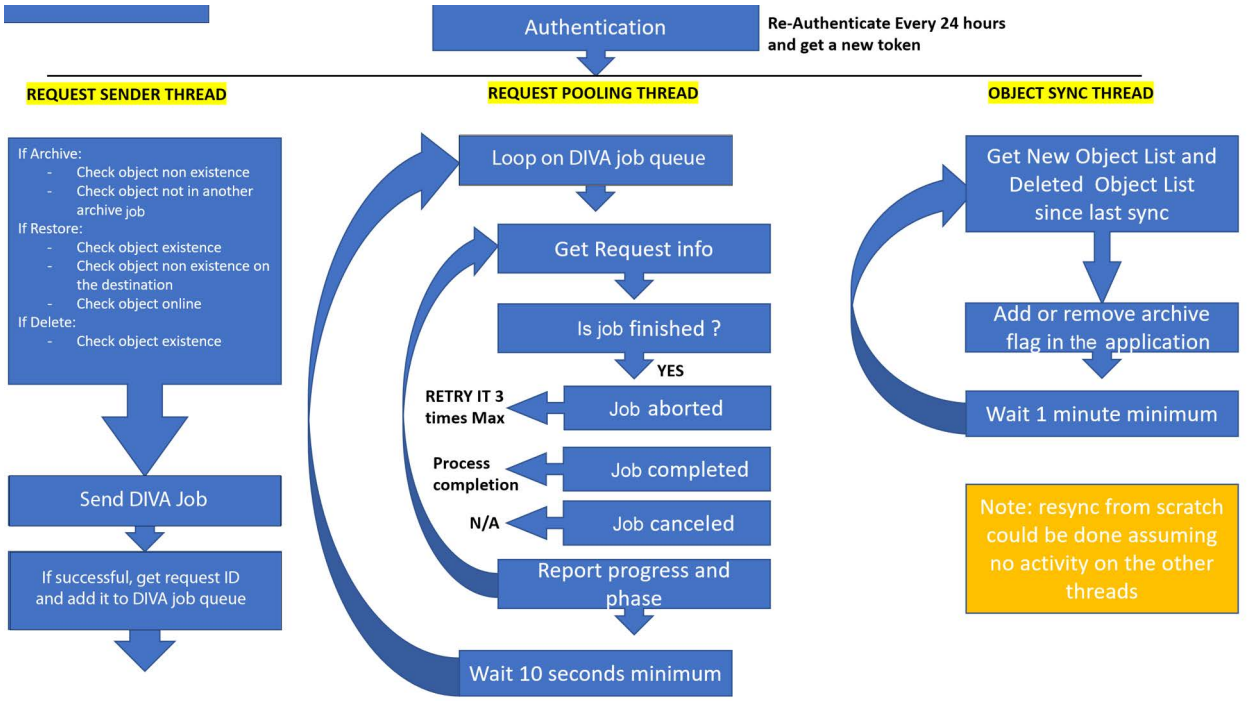
- `POST /users`
- `DELETE /users`
- `GET /users`
- `GET /roles`

DIVA REST API Workflow Guidelines

Use the following guidelines to develop workflows using the DIVA REST API:

- First authentication: if possible use only one authentication to DIVA at the start of the application and use the token returned for further API calls. Do not authenticate multiple times, and in particular not before each DIVA job.
- Send the DIVA job (Archive, Restore, and so on) using the token from the last step and get the Request ID. Add the Request ID to the DIVA job queue.
- Pool every `n` seconds on the DIVA job queue list using `getRequestInfo`. Wait a minimum of 10 seconds between each pooling phase.
- The progress and phase can be obtained for each running job.
- Any running job can be canceled.
- You can remove a finished job from the DIVA job queue. A finished job will be COMPLETED, PARTIALLY_COMPLETED, ABORTED, or CANCELLED.
- Avoid retrying too many times if a job fails.
- Before restoring an object, use `divaGetObjectInfo` to know if the object is online; there is no need to try to restore an offline object because it will fail.
- Try to develop a sync (or resync) mechanism to sync the application with DIVA objects using the Since Date option to discover new and deleted objects.

After authenticated, three different threads could be created to manage the DIVA workflows as shown in the following figure:



DIVA Job Status Codes

For descriptions of DIVA job status codes:, see *API Static Constant Values* on this Swagger page: Manager API.

This table identifies possible status codes for unsuccessful Archive requests:

Partial Restore Job Formats and Manager Responses

The following formats—each identified by an INT value in *format*—are used when issuing jobs to DIVA Manager:

- 0—Bytes (range)
- 1—Not Used
- 2—Video GXF (timecode)
- 3—Video SEA (timecode)
- 4—Video AVI MATROX (timecode)
- 5—Video MPEG2 TS (timecode)
- 6—Video MXF (timecode)
- 7—Video Pinnacle (timecode)
- 8—Video Omneon (timecode)
- 9—Video Leitch (timecode)

- 10—Video Quantel (timecode)
- 11—Autodetect which video format (timecode)
- 12—File/Folder Based
- 13—DPX (range)

Job and Response Sample

Here are Partial Restore jobs and Manager responses. Take note of the differences in offsets and formats.

Sample 1: Body for Bytes Partial Restore

```
{
  "destinationServer": "sourcedest",
  "minRequestPriority": -1,
  "instance": -1,
  "qos": 0,
  "offsets": [
    {
    "destinationFile": "DNxHD-mxf-wrap-conf.mov",
    "offsetPairs": [
      {
        "bytesEnd": 1,
        "bytesBegin": 0,
        "timeCode": false
      },
      {
        "bytesEnd": 2,
        "bytesBegin": 1,
        "timeCode": false
      }
    ],
    "sourceFile": "DNxHD-mxf-wrap-conf.mov"
    }
  ],
  "format": 0,
  "options": " ",
  "objectName": "Partial File",
  "maxRequestPriority": 100,
  "priority": -1,
  "filePathRoot": "restore",
  "collectionName": "Restore All Basic PFR",
  "destinationServer": "sourcedest",
  "minRequestPriority": -1,
  "instance": -1,
  "qos": 0,
  "offsets": [
    {
"offsets": [
  {
    "destinationFile": "DNxHD-mxf-wrap-conf.mov",
    "offsetPairs": [
      {
        "bytesEnd": 1,
```

```
        "bytesBegin": 0,
          "timeCode": false
      },
      {
        "bytesEnd": 2,
        "bytesBegin": 1,
        "timeCode": false
      }
    ],
    "sourceFile": "DNxHD-mxf-wrap-conf.mov"
  }
],
  "format": 0,
  "options": " ",
  "objectName": "Partial File",
  "maxRequestPriority": 100,
  "priority": -1,
  "filePathRoot": "restore",
  "collectionName": "Restore All Basic PFR"
  }
}
```

Sample 2: Body for Video GXF (timecode) Partial Restore

```
{
  "destinationServer": "sourcedest",
  "minRequestPriority": -1,
  "instance": -1,
  "qos": 0,
  "offsets": [
    {
      "destinationFile": "Profile.gxf",
      "offsetPairs": [
        {
          "timeCodeBegin": "00:00:00:00",
          "timeCodeEnd": "00:00:00:01",
          "bytesEnd": 0,
          "bytesBegin": 0,
          "timeCode": true
        }
      ],
      "sourceFile": "Profile.gxf"
    }
  ],
  "format": 2,
  "options": " ",
  "objectName": "Partial File",
  "maxRequestPriority": 100,
  "priority": -1,
  "filePathRoot": "restore",
  "collectionName": "Restore All Basic PFR"
}
```

Sample 3: Body for File-Folder based Partial Restore

```
{
  "destinationServer": "sourcedest",
```

```
  "minRequestPriority": -1,
  "instance": -1,
  "qos": 0,
  "offsets": [
    {
      "fileFolder": {
        "name": "DNxHD_mxf_wrap_conf.mov",
        "option": ""
      }
    },
    {
      "fileFolder": {
        "name": "test.mov",
        "option": ""
      }
    }
  ],
  "format": 12,
  "options": " ",
  "objectName": "Partial File",
  "maxRequestPriority": 100,
  "priority": -1,
  "filePathRoot": "restore",
  "collectionName": "Restore All Basic PFR"
}
```

Sample 4: Body for DPX (Range) PR

```
{
  "destinationServer": "sourcedest",
  "minRequestPriority": -1,
  "instance": -1,
  "qos": 0,
  "offsets": [
    {
      "range": {
        "end": 2,
        "begin": 1
      }
    },
    {
      "range": {
        "end": 4,
        "begin": 3
      }
    }
  ],
  "format": 13,
  "options": " ",
  "objectName": "Partial File",
  "maxRequestPriority": 100,
  "priority": -1,
  "filePathRoot": "restore",
  "collectionName": "Restore All Basic PFR"
```

Python DIVAScript

This chapter describes DIVAScript for Python via the DIVA REST API.

Topics

- Python DIVAScript
- Running DIVAScript in Server Mode
- DIVAScript Command-Line Mode Requirements

Python DIVAScript

DivaScript was developed in python. DIVAScript supports sending commands via the DIVA REST API. DIVAScript helps develop and execute automated functional tests via the DIVA REST API.

DivaScript supports most DIVArchive commands that are available via the DIVA REST API.

DIVAScript Commands Supported by DIVA REST API

- `logIn`
- `logOut`
- `addgroup`
- `objectinfo`
- `objlist`
- `reqinfo`
- `version`
- `archive`
- `restore`
- `partialrestore`
- `copy`

- `copyas`
- `delete`
- `trarchive`
- `scan`
- `stage`

DIVAScript Configuration File Settings

```
diva_serv: serve_ip
diva_port: 8765
diva_user: diva_user
diva_pass: diva_pass
protocol: https
diva_version: 9
```

Running DIVAScript in Server Mode

To run DIVAScript in server mode, do the following:

1. Open the Windows command-line interface.
2. Type `DIVAScript.exe` and press Enter.

   DIVAScript displays the list of valid DIVAScript terms. DIVAScript doesn't require a parameter.

DIVAScript Command-Line Mode Requirements

The requirements for running DIVAScript in Command-Line Mode follow:

You must first call the logIn with correct credentials so that the token is saved and used for next commands.

If a parameter value contains spaces or other special characters, place the value inside " "

Use the pipe (|) delimiter for terms that allow a list of values (e.g: filelist from archive or in / out from partialRestore).

Command line syntax is DivaScript.exe following commandName: one of the recognized command names, like `archive` or `delete`.

Other terms: depending on the command, additional terms may be required.

The command name must be the first parameter. All other terms can be placed anywhere in the command line.

DivaScript Command Syntax

logIn

Login a user, returns a created token

Synopsis

```
DivaScriptPy.exe logIn -u USERNAME -p PASSWORD
```

logOut

Logout a user, returns the status.

Synopsis

```
DivaScriptPy.exe logOut -u USERNAME
```

addgroup

Create a tape group in the configuration.

Synopsis

```
DivaScriptPy.exe addgroup [-aqos ARCHIVEQOS] [-cpr COMPRES-
SIONENABLED] [-cmt DESCRIPTION] [-en ENCRYPTIONENABLED] [-fmt
FORMAT] -grp NAME
   [-nrt NUMBEROFTAPESUSEDFORREPACK] [-prio PRIORITY] [-rqos
RESTOREQOS] [-set SET] [-repk USEDFORREPACK] [-vw VERIFYWRITE]
[-wf WORSTFITENABLED]
```

objinfo

Gets object information for a specific object.

Synopsis

```
DivaScriptPy.exe objinfo -obj OBJECT -col COLLECTION
```

objectsList

Gets object information for all objects based on filter criteria. Returns a list of objects. On success, print one line for each matching object (name@category).

Synopsis

```
DivaScriptPy.exe objlist [-time INITIALTIME] [-lt LISTTYPE] [-
objlt OBJECTSLISTTYPE] [-maxs MAXLISTSIZE] -obj OBJECTNAME -col
COLLECTIONNAME [-grp MEDIANAME] [-lod LEVELOFDETAIL]
```

reqinfo

Get request by ID. Return the current status of a DIVA request.

On success, print the current status of the request (first line is the numerical status; second line is the text status in capital letters).

If this command is issued with the *wait* parameter, the getRequestInfo command will continue to run until the request related to the entered Request ID completes.

This parameter is important when it is necessary to wait for one DIVA request to complete before another is submitted via DivaScript.

Synopsis

```
DivaScriptPy.exe reqinfo -req REQUESTID  [-wait WAIT]
```

version

Gets DIVA build version and API version. Returns the DIVA build version and the API version.

Synopsis

```
DivaScriptPy.exe version
```

archive

Submits an archive request to the DIVA Core Manager. This call returns as soon as the Manager accepts the request.

Use the `reqinfo` command to know when the request has been executed.

On success, divascript prints the request ID on the output.

Synopsis

```
Divascript.exe  archive -obj OBJECT -col COLLECTION -grp
MEDIA_GROUP -src SERVER -fpr FILES_PATH_ROOT -filelist FILELIST
[-com COMMENTS]
```

The following arguments are required: `-obj/--object, -col/--collection, -grp/--media_group, -src/--server, -fpr/--files_path_root, -filelist/--filelist`

When there multiple files, Use the pipe delimiter: `-filelist "test.mov|gxf-sample.gxf"`

```
DivaScriptPy.exe archive -col COLLECTIONNAME [-com COMMENTS] -
filelist COMPONENTS -fpr FILEPATHROOT -grp MEDIA -obj OBJECT-
NAME [-opt OPTIONS] [-prio PRIORITY] [-qos QOS] -src SOURCE-
SERVER archive
```

restore

Submits an Object Restore request to the Manager and the Manager chooses the appropriate instance to be restored. If the requested object is on media that is not available, the request will fail.

Use the `reqinfo` command to know when the request has been executed.

On success, divascript prints the request ID on the output

Synopsis

```
DivaScriptPy.exe restore -col COLLECTIONNAME -src DESTINATION-
SERVER -fpr FILEPATHROOT [-inst INSTANCE] -obj OBJECTNAME [-opt
OPTIONS] [-prio PRIORITY] [-qos QOS]
```

partialrestore

Submits a Partial Object Restore request to the Manager and the Manager chooses the appropriate instance to be restored. If the request was not accepted (for example, if the requested object is on media not currently available) the request will generate an error. The Manager will use the instanceID field to select the instance of the object to use for

the Partial Restore operation. The Manager will choose an appropriate instance to restore if -1 is used.

Use the `reqinfo` command to know when the request has been executed.

On success, divascript prints the request ID on the output

Synopsis

```
DivaScriptPy.exe partialrestore -obj OBJECTNAME -col COLLEC-
TIONNAME -src DESTINATIONSERVER -fpr FILEPATHROOT -fmt {Files
and Folders,Byte Offset,Timecode,Frames,DPX} [-in SOURCEFILE]
[-out DESTINATIONFILE] [-start START] [-end END] [-inst
INSTANCE]
```

When there is a list of values, Use the pipe delimiter: `-start "10|20|30" -end "100|200|300"`

copy

Submits a New Instance Creation request on the media specified by mediaName to the Manager, and the Manager chooses the appropriate instance to be created. The request will fail if the requested object is on media that is not available.

Use the `reqinfo` command to know when the request has been executed.

On success, divascript prints the request ID on the output

Synopsis

```
DivaScriptPy.exe copy -col COLLECTION [-inst INSTANCE] -grp
MEDIA -obj OBJECTNAME [-opt OPTIONS] [-prio PRIORITY]
```

copyas

Submits a request for copying an archived object to a new object, with another name or category, to the DIVA Core Manager. The Manager chooses the appropriate instance as the source of the copy. The request will fail if the requested object is on an unavailable media. All types of transfers (disk to disk, disk to tape, tape to disk, and tape to tape) are supported.

Use the reqinfo command to know when the request has been executed.

On success, divascript prints the request ID on the output

Synopsis

```
DivaScriptPy.exe copyas -col COLLECTION [-com COMMENTS] [-inst
INSTANCE] [-media MEDIA] -newcol NEWCOLLECTIONNAME -newmedia
NEWMEDIA -newobj NEWOBJECTNAME -obj OBJECT [-opt OPTIONS] [-
prio PRIORITY]
```

delete

Submits an Object Delete Request to the DIVA Core Manager. The Manager deletes the specified instance OR every instance of the object if the instance is -1.

Use the `reqinfo` command to know when the request has been executed.

On success, divascript prints the request ID on the output

Synopsis

```
DivaScriptPy.exe delete [-col COLLECTION] [-inst INSTANCE] [-
grp MEDIA] -obj OBJECTNAME [-opt OPTIONS] [-prio PRIORITY]
```

trarchive

Submits a TranscodeArchive request to the DIVA Core Manager. The original object will be restored to the local Actor cache, then transcoded to the format defined in the option field. A new object containing the transcoded clip will then be archived back to DIVA Core.

Use the `reqinfo` command to know when the request has been executed.

On success, divascript prints the request ID on the output.

Synopsis

```
DivaScriptPy.exe trarchive [-delc CASCADEDELETE] [-com COM-
MENTS] -dcol DESTINATIONCOLLECTIONNAME -dobj DESTINATIONOBJECT-
NAME -media MEDIA [-opt OPTIONS] [-prio PRIORITY] [-qos QOS] -
col SOURCECOLLECTIONNAME [-inst SOURCEINSTANCEID] -obj SOURCEO-
BJECTNAME
```

scan

Submits a scan tape request to the DIVA Core Manager. This call returns as soon as the Manager accepts the request.

Use the `reqinfo` command to know when the request has been executed.

On success, divascript prints the request ID on the output.

Synopsis

```
DivaScriptPy.exe scan -col COLLECTION [-com COMMENTS] [-exl
EXCLUSION] [-imm IMPORTMODE] [-incl INCLUSION] [-min MINFILE-
SIZE] [-prio PRIORITY] -barcode BARCODE -grp MEDIA -trial
TRIALMODE
```

stage

Submits a Stage request to the media specified by targetMedia.

Use the `reqinfo` command to know when the request has been executed.

On success, DivaScriptPy prints the request ID on the output.

Synopsis

```
DivaScriptPy.exe stage -col COLLECTION -days NUMDAYS -obj
OBJECT -opt OPTIONS [-prio PRIORITY] -retier RESTORETIER -sto-
cls STORAGECLASS -target TARGETMEDIA
```