Report automation example

This section provides a complete example of how to automate report creation with email delivery using the License Statistics API and common script languages for Windows and Linux.

Scenario

For this example, let's assume we want to create a Usage Per User report and deliver it by email in PDF format. The following additional report options are included:

Parameter	Value
API Endpoint	/api/v3/license-server/(server_id)/usage-per-user/(output_format)
Date Range	Current Year
License Server Id	15
Aggregation Type	User and Host
Granularity	Day
Offset	0
Limit	20
Order	Descending by Date

Scripts

This scenario includes two example scripts, attached, which you can use as a basis for your own scripts.

For each script, adjust the following sections to suit your needs.

Section	Property	Description	PowerShell example (report.ps1)	Bash example (report.sh)
Configuration				
License Statistics	licstat_host	URL to the License Statistics server	$licstat_host = "https://licstat.example.com"	licstat_host=https://licstat.example.com
	api_token	API token to authorize requests	$api_token = "63675f08c441516f3ac4370deb28e01b"	api_token=63675f08c441516f3ac4370deb28e01b
Date Range You can use custom values or one of the pre-defined values in the script, including: • CURRENT YEAR	start_date	Start date of the report	$start_date = Get-Date -Format "yyyy-01-01"	start_date=$(date +%Y-01-01)

• LAST YEAR • CURRENT QUARTER • LAST QUARTER • CURRENT MONTH • LAST MONTH • CURRENT WEEK • LAST WEEK • YESTERDAY	end_date	End date of the report	$end_date = Get-Date -Format "yyyy-MM-dd"	end_date=$(date --iso-8601=date)
Report All properties depend on the specific reports and their requirements; this is just an example for the scenario. Adjust the parameters and their values according to your needs.	server_id	License Server Id	$server_id = 15	server_id=15
	aggregation_type	USERNAME, HOSTNAME, USER_HOST, USERGROUP, HOSTGROUP	$aggregation_type = "USER_HOST"	aggregation_type=USER_HOST
	granularity	HOUR, DAY, WEEK, MONTH, QUARTER, YEAR	$granularity = "DAY"	granularity=DAY
	offset	How many rows to skip from the beginning	$offset = 0	offset=0
	limit	Maximum number of rows to return	$limit = 20	limit=20
	order	String representation of column ordering	$order = '[{"property":"fud","direction":"DESC"}]'	order='[{"property":"fud","direction":"DESC"}]'
	fields_visibility	List of fields to be included	$fields_visibility = ('{"fud":1,"ugn":0,"hgn":0,"un":1,"uid":0,"hn":1,"hid":0,' + '"hip":0,"lsn":1,"lsid":0,"lsd":0,"fns":1,"fv":1,"ftype":0,"fcol":0,"fid":0' + ',"fd":0,"musage":1,"hu":1,"minu":0,"au":0,"mu":1,"hb":0,"minb":0,"mb":0,"ldtc":1,"ft":1}')	fields_visibility='{"fud":1,"ugn":0,"hgn":0,"un":1,"uid":0,"hn":1,"hid":0,"hip":0, "lsn":1,"lsid":0,"lsd":0,"fns":1,"fv":1,"ftype":0,"fcol":0,"fid":0,"fd":0, "musage":1,"hu":1,"minu":0,"au":0,"mu":1,"hb":0,"minb":0,"mb":0,"ldtc":1,"ft":1}'
Output	output_format	JSON, CSV, XLSX, PDF	$output_format = "PDF"	output_format=PDF
	output_filename	Filename to save the report to	$output_filename = "report"	output_filename=report
Download Report	report_file	Report filename format	$report_file = "$($output_filename).$($output_format.ToLower())"	report_file="${output_filename}.$(echo ${output_format} | tr '[:upper:]' '[:lower:]')"

	request_parameters	List of request parameters including specific API endpoint and its requirements	curl \
			--silent \
			--fail \
			-X POST \
			--location "${licstat_host}/api/v3/license-server/${server_id}/usage-per-user/${output_format}" \
			--header "X-Auth-token: ${api_token}" \
			--data-urlencode "lsid=${server_id}" \
			--data-urlencode "sd=${start_date}" \
			--data-urlencode "ed=${end_date}" \
			--data-urlencode "grat=${granularity}" \
			--data-urlencode "agrt=${aggregation_type}" \
			--data-urlencode "offset=${offset}" \
			--data-urlencode "limit=${limit}" \
			--data-urlencode "order=${order}" \
			--data-urlencode "fieldsVisibility=${fields_visibility}" \
			--output "${report_file}" || exit 1
				$request_parameters = @{
				Uri = "$($licstat_host)/api/v3/license-server/$($server_id)/usage-per-user/$($output_format)"
				Headers = @{"X-Auth-token" = $api_token}
				OutFile = $report_file
				Method = "POST"
				Body = @{
				lsid = $server_id
				sd = $start_date
				ed = $end_date
				grat = $granularity
				agrt = $aggregation_type
				offset = $offset
				limit = $limit
				order = $order
				fieldsVisibility = $fields_visibility
				}
				}

Send Report by email	subject	Email subject		
	recipient	Recipient email address		
	body	Email body		
	from	Sender email address		
	smtp_server	SMTP server address		

Executing the script

PowerShell

To run the script from the console, enter the following:

```
PS> .\report.ps1
```

To run the script from the command line, enter the following:

```
powershell.exe -File report.ps1
```

Bash

```
$ ./report.sh
```

Executing the script periodically

You can execute the script periodically using Windows Task Scheduler or Linux CRON.