Intel® Deep Learning Studio

Command Line Interface (CLI) Commands 1.0

Revision 1.0, July 2018

Intel Confidential—Do Not Reproduce

Terms and Conditions

This document contains information on products and/or processes in development.

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL PRODUCTS AND/OR SERVICES. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN AN EXECUTED AGREEMENT, INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS AND SERVICES INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT, OR OTHER INTELLECTUAL PROPERTY RIGHT.

UNLESS OTHERWISE AGREED IN A SIGNED WRITING BY INTEL, THE INTEL PRODUCTS AND SERVICES ARE NOT DESIGNED NOR INTENDED FOR ANY APPLICATION IN WHICH THE FAILURE OF THE INTEL PRODUCT COULD CREATE A SITUATION WHERE PERSONAL INJURY OR DEATH MAY OCCUR.

Intel may make changes to specifications and product /process descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

The products and services described in this document may contain defects or errors known as errata which may cause deviations from published specifications. Current characterized errata are available on request.

Document Revision History

Revision	Date	Comments
1.0	July 2018	Initial version.

1 DLS4E Command Line Interface (CLI) Commands

This document lists all DLS4E CLI commands, subcommands, options, and arguments. Precede the following commands with **dlsctl**. Please see each *subcommand* following this summary below.

Command	Description
experiment, exp	Command for submitting and managing training experiments and launching with Jupyter.
launch, l	Command for launching web user-interface or tensorboard.
mounts, m	Command displays a command that should be used to mount client's folders on his/her local machine.
user, u	Command for creating/deleting/listing users of the platform. Can be only run by a platform administrator.
verify, ver	Command used to verify whether all external components required by dlsctl are installed in proper versions. If something is missing - application displays a detailed information about it.
version, v	Displays version of dlsctl application.
-h, --help	Shows help for commands and arguments.

1.1 dlsctl experiment command and subcommands

1.1.1 dlsctl experiment command

Purpose	This overall purpose of this command/subcommands is to manage experiments.
Syntax	dlsctl exp [Options] [subcommand] [Arguments]...
Options - See each subcommand.	
--help -h	Show this information and exit.
Subcommands	

submit, s	Submitting training scripts for a single or multi-node tensorflow training.
interact, i	Launches a local browser with Jupyter notebook. If script name argument is given, that script is put into opened notebook. (--no_launch option also starts Jupyter session without starting a browser.)
view, v	Displays details of experiment with a given name.
list, ls	List experiments. Lists names, parameters used, result metrics, submission date, user, and status of jobs submitted.
cancel, c	Cancels experiment/s chosen based on criteria given as an argument.
logs, lg	Show logs for given experiment.
template_list, t	Command returns a list of available templates that can be used to submit training jobs.

1.1.2 dlsctl experiment submit subcommand

Purpose	Submit experiments.
Syntax	dlsctl experiment submit [Options] SCRIPT_LOCATION [SCRIPT_PARAMETERS]... Additional files relevant to the experiment
Script_folder_location, -sfl PATH	--script_folder_location Name of a folder and subfolders with additional files used by a script. Examples include other .py files, data, etc. If not given - its content won't be copied into an image.
Options	
--template, -t	Name of a template used to create a deployment. By default a single-node tensorflow training template is chosen. List of available templates might be obtained by issuing dlsctl experiment template_list command.
--name, n	Name of this experiment.

--pack_param, -p	Additional pack param in format: 'key value' or 'key.subkey.subkey2 value' For lists use: 'key "['val1', 'val2']"' For maps use: 'key "{'a': 'b'}"'
--parameter_range, -pr	Start multiple experiments using a range: - range - *{x...y:step}* - launch a number of experiments using values between *x* and *y* (including both values) with step *step*. -pr parameter gives also a possibility to pass explicit values not only values calculated based on start/stop/step - i.e. {x, y, z}
--parameter_set, -ps	Launch an experiment with a set of parameters defined in argument. Format of the argument is as follows : *{<param1_name>: <param1_value>, <param2_name>: <param2_value>, ..., <paramn_name>:<paramn_value>}*.
--verbose, -v	Set verbosity level: • -v for INFO • -vv for DEBUG. Provides additional information to help resolve a problem.
--help, -h	Show this information and exit.
SCRIPT_PARAMETERS	Parameters that are passed to the training script should be placed after "--" Example: dlsctl experiment submit <script_name.py> -- -script-parameter 1 ...

1.1.3 dlsctl experiment interact subcommand

Purpose	Opens an existing interactive session with Jupyter Notebook or creates a new one.
Syntax	dlsctl experiment interact [Options]
Options	
--name, -n	This is the Jupyter notebook session name. If a session exists with that name, it will be opened. If not, a new session will be created.
--filename, -f	File with a notebook that should be opened in Jupyter notebook.
-p, --	Additional pack param in format: 'key value' or 'key.subkey.subkey2 value'

pack_param.	For lists use: 'key "['val1', 'val2']"' For maps use: 'key "{'a': 'b'}"'
--no-launch	Run command without a web browser starting; only proxy tunnel is created.
-p, --port	Port on which service will be exposed locally.
--verbose, -v	Set verbosity level: • -v for INFO • -vv for DEBUG. Provides additional information to help resolve a problem.
--help, -h	Show this information and exit.

1.1.4 dlsctl experiment view subcommand

Purpose	Displays details of an experiment.
Syntax	dlsctl experiment view [Options] EXPERIMENT_NAME
Required	EXPERIMENT_NAME is the experiment name, given when the experiment is first submitted (launched).
Options	
--tensorboard, -tb	If given, exposes a tensorboard's instance with an experiment's data.
--verbose, -v	Set verbosity level: • -v for INFO • -vv for DEBUG. Provides additional information to help resolve a problem.
--help, -h	Show this information and exit.

1.1.5 dlsctl experiment list subcommand

Purpose	Lists experiments.
Syntax	dlsctl experiment list [Options]
Options	

--all-users, -a	Show all experiments, regardless of owner.
--name, -n	A regular expression to create a list of experiments for which the names match this expression.
--status, -s	[QUEUED|CREATING|RUNNING|COMPLETE|CANCELLED|FAILED]
--verbose, -v	Set verbosity level: • -v for INFO • -vv for DEBUG. Provides additional information to help resolve a problem.
--help, h	Show this message and exit.

1.1.6 dlsctl experiment cancel subcommand

Purpose	Cancels experiments currently in progress or completed.
Syntax	dlsctl experiment cancel [OPTIONS] EXPERIMENT_NAME
Required	EXPERIMENT_NAME is the name of the experiment or experiments
Options	
--match -m	A regular expression to create a list of experiments for which the names match this expression. If this option is used, then there is no need to pass EXPERIMENT_NAME
--purge, -p	If given - all information concerning experiments is removed from the system.
--verbose, -v	Set verbosity level: • -v for INFO • -vv for DEBUG. Provides additional information to help resolve a problem.
--help, -h	Show this information and exit.

1.1.7 dlsctl experiment logs subcommand

Purpose	Show logs for given experiment name.

Syntax	dlsctl exp logs [Options] EXPERIMENT_NAME
Required	Name of experiment for which logs will be retrieved.
Options	
--min-severity, -s	[CRITICAL|ERROR|WARNING|INFO|DEBUG] Show all events with this specified minimal and greater severity.
--start-date, -sd	Retrieve logs produced from this date, compliant with the ISO 8601 format. For example: 2007-04-05T24:00:00.
--end-date, -ed	Retrieve logs produced until this date, compliant with the ISO 8601 format. For example: 2007-04-06T24:00:00.
--pod-ids, -i	Comma separated list of pod IDs, if provided, only logs from these pods will be returned.
--pod-status, -p	[PENDING|RUNNING|SUCCEEDED|FAILED|UNKNOWN] Get logs only for pods with given status.
--verbose, -v	Set verbosity level: -v for INFO -vv for DEBUG
--help, -h	Show this information and exit.

1.1.8 dlsctl experiment template_list subcommand

Purpose	This command lists available draft templates for running experiments.
Syntax	dlsctl experiment template_list [Options]
Options	

--help, -h	Show this information and exit.

1.2 dlsctl launch command and subcommands

1.2.1 dlsctl launch command

Purpose	Commands to launch tensorboard or launch the Web UI.
Syntax	dlsctl launch [Options] [subcommand]
Subcommands	
tensorboard, tb	Subcommand for launching tensorboard.
webui, ui	Subcommand for launching webUI.
Options: See each subcommand.	
-h --help	Show this information and exit.

1.2.2 dlsctl launch tensorboard

Purpose	Subcommand for launching tensorboard with credentials
Syntax	dlsctl launch tensorboard, tb [Options] EXPERIMENT_NAME
EXPERIMENT_NAME	Name of experiment.
Options	
--verbose, -v	Set verbosity level: • -v for INFO • -vv for DEBUG. Provides additional information to help resolve a problem.
--no-launch, -n	Run command without a web browser starting, only proxy tunnel is created.
--help, -h	Show this information and exit.

1.2.3 dlsctl launch webui

Purpose	Subcommand for launching web UI local browser with credentials.
Syntax	dlsctl launch webui [Options]
Options	
--no-launch, -n	Run command without a web browser starting, only proxy tunnel is created.
--verbose, -v,	Set verbosity level: • -v for INFO • -vv for DEBUG. Provides additional information to help resolve a problem.
--help, -h	Show this information and exit.

1.3 dlsctl mounts command

Purpose	Command displays a command that should be used to mount a client's folders on his/her local machine.
Syntax	dlsctl mounts, m [Options]
Options	
--verbose, -v	Set verbosity level: • -v for INFO • -vv for DEBUG. Provides additional information to help resolve a problem.
--help, -h	Show this information and exit.

1.4 dlsctl user command and subcommands

1.4.1 dlsctl user command

Purpose	Command manages user accounts.
Syntax	dlsctl user [Options] COMMAND [Argument]...
Argument	USERNAME

Subcommands	
create, c	Command creates a new user on the platform. Can be only run by a platform administrator.
delete, d	Command deletes a user from the platform. Can be only run by a platform administrator.
list, ls	Lists users
Options -See each subcommand.	
--help, -h	Show this information and exit.

1.4.2 dlsctl user create subcommand

Purpose	Creates a new user with a name given as an argument. Can be only run by a platform administrator.
Syntax	dlsctl user create [Options] USERNAME
USERNAME Required	Name of user to be created.
Options	
--verbose, -v	Set verbosity level: • -v for INFO • -vv for DEBUG. Provides additional information to help resolve a problem.
--help, -h	Show this information and exit.

1.4.3 dlsctl user list subcommand

Purpose	Lists all users.
Syntax	dlsctl user list [Options]
Options	
--verbose, -v	Set verbosity level: • -v for INFO • -vv for DEBUG. Provides additional information to help resolve a

	problem.
--help -h	Show this information and exit.

1.4.4 dlsctl user delete subcommand

Purpose	Deletes a user with a name given as a parameter.
Syntax	dlsctl user delete [Options] USERNAME
Required	USERNAME
Options	
--purge, -p	If this option is added - command removes all client's artifacts.
--verbose, -v	Set verbosity level: • -v for INFO • -vv for DEBUG. Provides additional information to help resolve a problem.
--help, -h	Show this information and exit.

1.5 dlsctl verify command

Purpose	Verifies that all required subcomponents are installed, with required versions, for example, draft, kubectl,docker, etc.
Syntax	dlsctl verify [Options]
Options	
--verbose, -v	Set verbosity level: • -v for INFO • -vv for DEBUG. Provides additional information to help resolve a problem
--help, -h	Show this information and exit.

1.6 dlsctl version command

Purpose	Returns the version of the dlsctl application and the version of the platform.
Syntax	dlsctl version [Options]
Options	
--verbose, -v	Set verbosity level: • -v for INFO • -vv for DEBUG. Provides additional information to help resolve a problem
--help, -h	Show this information and exit.