Skyline Command-Line Interface

The command-line interface for Skyline is called SkylineRunner.exe. It is a tiny shim executable less than 10 KB in size. It requires a full Skyline installation on the computer on which it is run. SkylineRunner simply starts Skyline running without any user interface, pipes the parameter options from the command-line to the running Skyline instance, and prints output from Skyline to the command console. At present, only one instance of SkylineRunner may be executed at a time.

The Skyline instance started by SkylineRunner is independent of any other instances that may already be running on the same machine. It is not necessary to have a visible instance of Skyline running on your computer for SkylineRunner to work.

SkylineRunner is intended for automating tasks, such as quality control, scheduling and refinement, during acquisition. SkylineRunner can open a Skyline document, import a newly acquired data file, and export a report or new method.

# Parameter Options:

The current implementation of SkylineRunner offers the following options:

## General input/output

|  |  |
| --- | --- |
| --in=path/to/file.sky | Open a Skyline file |
| --save | Saves any changes to the file |
| --out=path/to/file.sky | Same as save except writes to the specified file |
| --batch-commands =path/to/file | Runs a file line by line treating each line like a SkylineRunner input. Useful for automating the execution of multiple commands. The open Skyline file remains active through all commands. |

*Until the section titled Settings Customization all other command line parameters rely on the “in” parameter because they all rely on having a Skyline document open.*

## Importing results replicates

|  |  |
| --- | --- |
| --import-file=path/to/file | Attach a replicate to the open document |
| --import-replicate-name=<name> | Name to give the new replicate in an –import-file operation. |
| --import-optimizing=<ce | dp> | Indicates the data being imported contains extra transitions for detecting optimal collision energy or declustering potential. |
| --import-append | Append the import-file to the given replicate. This is an intention check in case the document already has a replicate with the given name. By default this is set to false. This option only works with the –import-file option. |
| --import-all=path/to/folder | Imports from a folder all files or sub-folders which are not already in the document, naming each with the base-name of the file or sub-folder, unless the --import-naming-pattern parameter is also supplied. |
| --import-naming-pattern=reg-ex | A regular expression from which the first group will be used to name replicates in an --import-all operation (e.g. [^\_]\_(.\*) for everything after the first underscore). |
| --import-before=<date> | When importing from a folder, only import from files with modified time before the given date. |
| --import-on-or-after=<date> | When importing from a folder, only import from files with modified time after the given date. |

## Removing results replicates

|  |  |
| --- | --- |
| --remove-before=<date> | Remove all results from the open document with an acquired time before the given date. |

## Importing FASTA files

|  |  |
| --- | --- |
| --import-fasta=path/to/file | Import a FASTA file into the open document. |

## Adding spectral libraries

|  |  |
| --- | --- |
| --add-library-path=path/to/file | Specify a spectral library to be added to the open document. |
| --add-library-name=<name> | Name to give the spectral library in an –add-library-path operation. |

## Exporting reports

|  |  |
| --- | --- |
| --report-name=<name> | The name of a report to export as it appears in the Skyline Export Report form |
| --report-file=path/to/file.csv | The path to export the report to. Required if --report-name is specified. |
| --report-format=<CSV | TSV> | CSV for comma-separated reports (or semicolon separated, depending on your localization) or TSV for tab separated reports [default CSV] |

## Exporting transition lists

|  |  |
| --- | --- |
| --exp-translist-instrument=<AB Sciex | Agilent | Thermo | Waters> | Export a transition list. This option is required for exporting a transition list and has no default. This option cannot be used with --exp-method-instrument, because you cannot export a method and transition list simultaneously. |

## Vendor-specific transition list options

|  |  |  |
| --- | --- | --- |
| AB Sciex | --exp-dwell-time=<millis> | Dwell time per transition. This option is required for unscheduled transition lists. |
| Agilent | --exp-dwell-time=<millis> | Same as above. |
| Thermo Scientific | --exp-add-energy-ramp | Adds an extra column for energy ramp to the transition list. Optional. Defaults to false. |
| Waters | --exp-run-length=<minutes> | Run length of the entire gradient in minutes. This option is required for unscheduled experiments. |

## Exporting native instrument methods

|  |  |
| --- | --- |
| --exp-method-instrument=<AB SCIEX QTRAP | Agilent 6400 Series | Thermo TSQ | Thermo LTQ | Waters Xevo | Waters Quattro Premier> | Export a method. This option is required for exporting a method and has no default. This option cannot be used with --exp-translist-instrument, because you cannot export a method and transition list simultaneously. |
| --exp-template=path/to/file.meth|exp|dam|m | Path of the method template. This can be a file or a directory depending on your instrument. This option is required for method export. |

## Vendor-specific method options

|  |  |  |
| --- | --- | --- |
| AB Sciex Qtrap | --exp-dwell-time=<millis> | Dwell time per transition. This option is required for standard (unscheduled) methods. |
| Agilent (all instruments) | --exp-dwell-time=<millis> | Same as above |
| Thermo (all but LTQ) | --exp-run-length=<minutes> | Run length of the entire gradient in minutes. This option is required for unscheduled experiments. |
| Waters (all instruments) | --exp-run-length=<minutes> | Same as run length above |

## Method and transition list options

|  |  |
| --- | --- |
| --exp-file=path/to/file | Path to the method or transition list file (or directory) to export to. This option is required for method and transition list export. |
| --exp-strategy=<single | protein | buckets> | Strategy for dividing a method into injections. The default is “single”. |
| --exp-method-type=<standard | scheduled | triggered> | Sets a standard, scheduled or triggered method. The default is “standard”. |
| --exp-max-trans=<number> | Maximum number of transitions per injection for export strategies “protein” and “buckets” OR maximum number of simultaneous transitions for scheduled methods. The default is 100. |
| --exp-optimizing=<ce | dp> | Export a method with extra transitions for finding optimal collision energy or declustering potential. |
| --exp-scheduling-replicate=<name> | Use this only if creating a scheduled or triggered method. The default is to schedule based on an average of all replicates, but if you specify one, the method will be scheduled based on that replicate. |
| --exp-ignore-proteins | Ignore protein boundaries in creating methods. |
| --exp-primary-count=<number> | For --exp-method-type=triggered specifies the number of transitions to make primary. |

## Settings Customization

*The below commands do not rely on the “in” parameter because they modify the user settings that are independent of a specific Skyline document.*

|  |  |
| --- | --- |
| --tool-add=<name> | The name of a tool to add to the Tools menu in Skyline. To be added, a tool must have a name and command. If there is a name conflict the -- tool-conflict-resolution parameter is required. To add more than one tool at a time you may use the –batch-commands parameter below. |
| --tool-command=path/to/file.exe  (also supports com, pif, bat, cmd)  --tool-command=http://server/file.html | The command or web URL of the tool to add to the Tools Menu. To be added, a tool must have a name and command. |
| --tool-arguments=”<arguments>” | Optional command-line arguments for the tool to be added, used when the tool is executed.  (Not applicable to web URL commands) |
| --tool-initial-dir=path/to/dir | Optional initial directory for the tool to be added, used when the tool is executed. (Not applicable to web URL commands) |
| --tool-conflict-resolution=<overwrite | skip> | Tells the SkylineRunner how to resolve a tool name conflict, by either overwriting an existing installation or skipping installation of the new tool. |
| --tool-report=<report-name> | The name of a report in the settings to use as the input report for the tool. |
| --tool-output-to-immediate-window | When present the tool output is piped to the Immediate Window at runtime. |
| --report-add=path/to/file.skyr | Adds the report formats from a skyr file. If there are name conflicts the --report-conflict-resolution parameter is required. |
| --report-conflict-resolution=<overwrite | skip> | Tells the SkylineRunner how to resolve a report name conflict, by either overwriting the existing report or skipping adding the new report. |
| --tool-add-zip=path/to/file.zip | Import tools from a tool installation ZIP file. |
| --tool-zip-conflict-resolution=<overwrite | parallel> | Specify whether tool conflicts from the provided ZIP file should be resolved by overwriting or installing in parallel. This is for conflicts related to tool versioning and report names. |
| --tool-zip-overwrite-annotations=<true | false> | Specify whether conflicting custom annotations from the provided ZIP file should overwrite (true) existing annotations or be skipped (false). |
| --tool-program-macro=<programTitle>  Or  --tool-program-macro=<programTitle>,<programVersion>  Eg.  --tool-program-macro=R,2.15.2 | Specifies a program title and version to use with the –tool-program-path command. Together these commands are for importing tools from a ZIP file that use the $(ProgramPath()) macro as their command. For more information see the documentation on External Tools. |
| --tool-program-path=path/to/file | Specifies the path to an executable on the local machine for the program title and version specified by the –tool-program-macro flag. |
| --tool-ignore-required-packages | Ignore required packages when installing a tool from a ZIP file. |