



## Command Line Guide

Updated March 15, 2019 for use with  
3DMark Professional Edition v2.8.6546

## Installation

When installing the application using a command line the following options are available.

3dmark-setup.exe [options]

Command	Description
/installpath=<install path>	Defines the install path, default is C:\Program Files\Futuremark\3DMark
/quiet /silent	Silent install, displays no user interface
/force	Force install
/install	Installs the product (Default)
/uninstall	Uninstalls the product



Running the installer while using elevated permissions can cause the application to not function correctly. Please avoid running the installer with elevated permissions until prompted.

## Usage

Run the program from a command line that was started as an administrator, (right-click on the *cmd* shortcut, and select *Run as Administrator*).

3DMarkCmd.exe [options]



[3DMark Professional Edition](#) license required for command line use.

## Options



When *on/off* is omitted with an option, *on* is assumed.

Command	Description
<code>--runall</code>	Run all workloads, including Fire Strike Extreme, Ice Storm Extreme and Ice Storm Unlimited. Demos are skipped. Alternative to <code>--definition=</code> .
<code>--definition=&lt;benchmark.xml&gt;</code>	Name of benchmark definition XML file. Specifies the XML file that defines the tests and settings to be used. See below for XML file example.
<code>--loop[=&lt;count&gt;]</code>	Set the number of times to loop benchmark. The default is 1. Use 0 for infinite loop stress test, benchmark will not end until aborted.
<code>--audio[=on =off]</code>	Play audio (default on).
<code>--systeminfo[=on =off]</code>	Collect SystemInfo (default off).
<code>--systeminfomonitor[=on =off]</code>	Enable SystemInfo Monitoring (default off).
<code>--out=&lt;file.3dmark-result&gt;</code>	Save results in the file.
<code>--in=&lt;file.3dmark-result&gt;</code>	Load results from the file (only for submitting online using <code>--online=on</code> or exporting using <code>--export</code> )
<code>--online[=on =off]</code>	Send results to Futuremark Online (default off).
<code>--export=&lt;file.xml&gt;</code>	Export results to the XML file.
<code>--feature_level=&lt;level&gt;</code>	Specify DirectX feature level, which can be one of: 9_1, 9_2, 9_3, 10_0, 10_1 or 11_0. Note that if you limit this and attempt to run tests that require higher feature level, you will get an error that your hardware doesn't support the required features.

<code>--scalingmode=&lt;mode&gt;</code>	Specify scaling mode, which can be one of: centered, stretched.
<code>--log=&lt;log-file&gt;</code>	Save benchmark progress log to <code>&lt;log-file&gt;</code> . Logging does not affect scores.  If this option is not used, the last 1000 lines of logging are saved to the default location: C:\Users\*username*\Documents\3DMark\Log\3DMark.log
<code>--debug-log</code>	Enable per workload debug logging. Log files for each workload run are saved to: C:\Users\*username*\Documents\3DMark\Logs
<code>--register=&lt;product key&gt;</code>	Register 3DMark with the given key.
<code>--unregister</code>	Unregister 3DMark.
<code>--install=&lt;DLC file path&gt;</code>	If you have a standalone 3DMark dlc file it can be installed by providing the path to the DLC file
<code>--uninstall</code>	Removes all DLCs and settings files
<code>--path=&lt;arg&gt;</code>	Changes the destination path for downloaded DLCs and other runtime files
<code>--language=&lt;arg&gt;</code>	Changes the language of the application
<code>--encodedParameters=&lt;arg&gt;</code>	Passes command line parameters to the application in encoded form
<code>--trace</code>	Verbose logging



You may have problems running Time Spy, Time Spy Extreme and Night Raid from a PowerShell terminal due to the way PowerShell works with full-screen DirectX 12 applications. Use the following Command Line option to work around this issue:  
start /max 3DMarkCMD.exe

## Examples

These examples assume that you have mybenchmark.3dmdef in the /bin/x64 folder which defines your benchmark run and that you have write permissions to the same directory.

**Example:** Run all workloads (Ice Storm, Ice Storm Extreme, Ice Storm Unlimited, Cloud Gate, Sky Diver, Fire Strike and Fire Strike Extreme), save result file to myresults.3dmark-result

```
3DMarkCmd.exe --runall --out=myresults.3dmark-result
```

**Example:** Loop Fire Strike, including demo, indefinitely using default settings.

```
3DMarkCmd.exe --definition=..\..\firestrike.3dmdef --loop=0  
--out=myresults.3dmark-result
```

**Example:** Loop three times with customized "mybenchmark.3dmdef" settings, saving results to myresults.3dmark-result (there will be three numbered result files, one per run)

```
3DMarkCmd.exe --definition=mybenchmark.3dmdef --loop=3  
--out=myresults.3dmark-result
```

**Example:** Install DLCs downloaded separately.

```
3DMarkCmd.exe --install="C:\downloads\"
```

**Example:** Change language to German.

```
3DMarkCmd.exe --language=de-DE
```

**Example:** Change DLC install path

```
3DMarkCmd.exe --path="D:\3DMarkD1c"
```

## Definition XML files

3DMark comes with definition files that enable you to set up and run a benchmark with standard or custom settings. By default, these definitions can be found in:

C:\Program Files\UL\3DMark\

(Modify the directory to x86 instead of x64 if running 32 bit OS.)



Definition file names changed with the release of 3DMark v1.3.708, which added the Sky Diver test. You may need to update existing scripts.

apioverhead.3dmdef	Run default API Overhead test
icestorm.3dmdef	Run default Ice Storm test
icestorm_extreme.3dmdef	Run default Ice Storm Extreme test
icestorm_unlimited.3dmdef	Run default Ice Storm Unlimited test
cloudgate.3dmdef	Run default Cloud Gate test
skydiver.3dmdef	Run default Sky Diver test
firestrike.3dmdef	Run default Fire Strike test
firestrike_extreme.3dmdef	Run default Fire Strike Extreme test
firestrike_ultra.3dmdef	Run default Fire Strike Ultra test
timespy.3dmdef	Run default Time Spy test
timespy_extreme.3dmdef	Run default Time Spy Extreme test
nightraid.3dmdef	Run default Night Raid test
portroyal.3dmdef	Run default Port Royal test
run_all.3dmdef	Runs all the above tests in sequence <sup>1</sup>
stresstest_firestrike_performance.3dmdef	Run default Fire Strike Stress Test

<sup>1</sup> Fire Strike Ultra is not included in this definition since systems with less than the recommended 3 GB of video card memory may crash when trying to run the test.

stresstest_firestrike_extreme.3dmdef	Run default Fire Strike Extreme Stress Test
stresstest_firestrike_ultra.3dmdef	Run default Fire Strike Ultra Stress Test
stresstest_skydiver_performance.3dmdef	Run default Sky Diver Stress Test
stresstest_timespy_performance.3dmdef	Run default Time Spy Stress Test
stresstest_timespy_extreme.3dmdef	Run default Time Spy Extreme Stress Test
stresstest_nightraid_performance.3dmdef	Run default Night Raid Stress Test
stresstest_portroyal_performance.3dmdef	Run default Port Royal Stress Test
custom_icestorm.3dmdef	Run Ice Storm with custom settings
custom_icestorm_extreme.3dmdef	Run Ice Storm Extreme with custom settings
custom_icestorm_unlimited.3dmdef	Run Ice Storm Unlimited with custom settings
custom_cloudgate.3dmdef	Run Cloud Gate with custom settings
custom_skydiver.3dmdef	Run Sky Diver with custom settings
custom_firestrike.3dmdef	Run Fire Strike with custom settings
custom_firestrike_extreme.3dmdef	Run Fire Strike Extreme with custom settings
custom_firestrike_ultra.3dmdef	Run Fire Strike Ultra with custom settings
custom_timespy.3dmdef	Run Time Spy with custom settings
custom_timespy_extreme.3dmdef	Run Time Spy Extreme with custom settings
custom_nightraid.3dmdef	Run Night Raid with custom settings
custom_portroyal.3dmdef	Run Port Royal with custom settings

Default definitions are the same as running a test from the GUI.

Custom definition files mirror the options available on the Custom tab of the GUI. Copy the appropriate custom definition file and edit it to match your desired settings. Note that custom runs only produce sub-scores, never an overall score.



## Example

### timespy.3dmdf

```
<?xml version="1.0" encoding="utf-8"?>
<benchmark>
  <application_info>
    <selected_workloads>
      <selected_workload name="TimeSpyDemoP"/>
      <selected_workload name="TimeSpyGt1P"/>
      <selected_workload name="TimeSpyGt2P"/>
      <selected_workload name="TimeSpyCpuP"/>
    </selected_workloads>
  </application_info>
</benchmark>
```

Test names are fairly self-explanatory, for example "TimeSpyDemoP" is the demo, "TimeSpyGt1P" the Graphics test 1 and so on. Note that you will only get an overall score when all sub-tests are included.

If all you want is to specify which tests to run (for example, to skip the Demo), just make a copy of the appropriate definition file and edit the list of tests.

### stresstest\_firestrike\_performance.3dmdf

```
<?xml version="1.0" encoding="utf-8"?>
<benchmark>

  <application_info>
    <selected_workloads>
      <selected_workload name="FireStrikeGt1XST"/>
    </selected_workloads>
```

```
</application_info>

<settings>
  <setting>
    <name>loop_count</name>
    <value>20</value>
  </setting>
</settings>
</benchmark>
```

The stress test definitions function in a similar manner to other definition files but must also contain a valid loop count setting (between 2 and 5000)

## Changes to .3dmdf files from v1.3 to v1.4

### Setting command line options with .3dmdf files

Some command line settings can now be set within .3dmdf files, for both default and custom benchmark runs.

If the same setting is set from both the command line and the .3dmdf file, the value differing from the default value will be used.

For example, if `--debug-log=off` is specified on the command line, but .3dmdf file contains the code below, then the debug log will be enabled since it is the non-default setting.

```
<setting>
  <name>enable_debug_log</name>
  <value>1</value>
</setting>
```

The table below lists all settings that can be set within a .3dmdf file.

.3dmdf setting name	.3dmdf default value	command line switch (with default value)
enable_debug_log	0	--debug-log=off
swapchain_height	0 (auto)	
swapchain_width	0 (auto)	
enable_audio	1	--audio=on
enable_systeminfo_monitor	0	--systeminfomonitor=off
enable_systeminfo_collect	0	--systeminfo=off
gpu_count	0 (autodetect)	
scaling_mode	centered	--scalingmode=centered <sup>2</sup>

---

<sup>2</sup> Using the `--scalingmode` setting on the command line always overrides the value in .3dmdf file.

## Using custom settings

Custom workloads now have separate identifiers. For example, to run Fire Strike with non-default resolution or other settings:

```
<application_info>
  <selected_workloads>
    <selected_workload name="FireStrikeDemoC"/>
    <selected_workload name="FireStrikeGt1C"/>
```

Specifying the "C" version of each workload name is required for the custom settings to have effect. If another name (for example FireStrikePhysicsP) is used, all custom settings that could potentially affect the score will be ignored.

## Redundant 'preset' attribute removed

The now redundant 'preset' attribute has been removed.

### 3DMark v1.3.708

```
<application_info>
  <selected_workloads>
    <selected_workload name="FireStrikeDemoP" preset="default"/>
    <selected_workload name="FireStrikeGt1P" preset="default"/>
```

```
<application_info>
  <selected_workloads>
    <selected_workload name="FireStrikeDemoP" preset="custom"/>
    <selected_workload name="FireStrikeGt1P" preset="custom"/>
```

### 3DMark v1.4.775

```
<application_info>
  <selected_workloads>
    <selected_workload name="FireStrikeDemoP"/>
    <selected_workload name="FireStrikeGt1P"/>
```

```
<application_info>
  <selected_workloads>
    <selected_workload name="FireStrikeDemoC"/>
    <selected_workload name="FireStrikeGt1C"/>
```

## GPU count setting in .3dmdf now has effect

In 3DMark v1.3 the `gpu_count` setting in `.3dmdf` file had no effect. In most sample files it had value 1.

In 3DMark v1.4 the `gpu_count` setting works as expected. For the majority of cases, the value should be 0 or omitted to enable auto-detection of the number of GPUs.