


# Dirac-3S Technical Reference: Software Interface

## September 2026, Revision A

Quantum Computing Inc
quantumcomputinginc.com
(703) 436-2161

# Contents

1 Software Package Installation 5

2 Writing an Initial Program for Dirac-3 5

3 Understanding Device Status 9
3.1 System Status . . . 9
3.2 Submission/Results Statuses . . . 10

4 Remote Command Line Interface for Administrators 11
4.1 Remote login . . . 11
4.2 Changing password . . . 11
4.3 Changing the network configuration . . . 12

5 Related Technical References 13

6 Contacting Support 13

Copyright 2026 Quantum Computing Inc.

# Document Revision History

| Version | Date | Changes |
|---|---|---|
| 0.0.1 | August, 2026 | N/A |

Copyright 2026 Quantum Computing Inc.

# Notice and Disclaimer

This Technical Reference, including all text, images, diagrams, and other content contained herein, is the intellectual property of Quantum Computing Inc. All rights, including copyright, are reserved. No part of this User Guide may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language without the prior written permission of Quantum Computing Inc.

Any unauthorized use or reproduction of this Technical Reference violate copyright laws and international treaties. The contents of this Technical Reference are provided for informational purposes only and may be subject to change without notice. Quantum Computing Inc. assumes no responsibility for errors or omissions in this document or for any damages, whether direct, indirect, incidental, consequential, or otherwise, arising from the use or inability to use the information contained herein.

All trademarks, trade names, logos, and service marks mentioned in this User Guide belong to their respective owners and are hereby acknowledged.

The use of this Technical Reference is limited to the intended purpose of assisting users in the proper usage of the associated hardware and software. Any other use, including but not limited to commercial exploitation, distribution, or modification of the content, is strictly prohibited.

By accessing and using this Technical Reference, you acknowledge and agree to respect the copyright and intellectual property rights of Quantum Computing Inc. If you have any questions or requests regarding the use of this User Guide, please contact support@quantumcomputinginc.com

Copyright 2026 Quantum Computing Inc.

# 1 Software Package Installation

The Dirac-3S device will be delivered with a static IP address of 192.168.42.13/24. To connect to it, configure your laptop's Ethernet adapter with an IP address in the 192.168.42.x range, where x is an unused address on the same subnet (e.g. not 13, 1, or 0) and the mask to 255.255.255.0. Once these settings have been changed, you may plug the Ethernet cable into the jack, and then into the host computer. In order to verify that the Dirac server is accessible, users must utilize the system monitoring capabilities of the Python package eqc-direct. In order to install *eqc-direct* the latest available version from PyPi run:

```
pip install eqc-direct
```

The package supports Python version 3.9 and 3.10. Next, using the Python package follow the instructions in the documentation under the System Monitoring heading to check if your device is up and running. When the device is powered on it begins by entering a calibration process to determine if the system is functioning properly. Using the system monitoring functions, users can check to see when their device is available to begin solving problems.

# 2 Writing an Initial Program for Dirac-3

Here we share a basic problem submission example for Dirac-3. All interactions with the device are resolved using eqc-direct a Python package which utilizes gRPC messaging to interface with the device. The package provides utility functions such as acquiring a lock on the device, system monitoring, testing the health of the device, submitting a problem, and cancelling a currently processing problem submission. In order to illustrate a standard problem submission a simple polynomial problem will be utilized:

$$E(x_1, x_2, x_3) = -10x_1 - 10x_3 - 3.7x_2^2 - 5x_1x_3 \ , \tag{1}$$

The first step is to extract the polynomial coefficients and format polynomial variable indices for each term in the equation. The polynomial coefficients will be represented as a list as follows:

```
poly_coefs = [-10, -10, -3.7, -5]
```

The polynomial indices for the coefficients will be represented as follows in the same order as represented in the original equation:

```
poly_indices = [[0,1], [0,3], [2,2], [1,3]]
```

Note, that each individual polynomial index set has non-decreasing values moving from left to right such that for each set of indices:

$$[k_1, k_2, k_3, \ldots, k_{n-1}, k_n] \tag{2}$$

Copyright 2026 Quantum Computing Inc.

the following condition must hold:

$$k_1 \leq k_2 \leq k_3 \leq \cdots \leq k_{n-1} \leq k_n \tag{3}$$

This property guarantees the uniqueness of the submitted polynomial index set. Also, this sparse representation minimizes the amount of data needed to be transferred to the device to obtain a solution.

After formatting the problem for submission, the client can be used to submit the problem after acquiring an exclusive execution lock on the device. The inputs to the function used to submit problems, *solve_sum_constrained*, are as follows:

- *lock_id* - a UUID string with exclusive lock for device execution
- *poly_coefficients* - the coefficient values for polynomial to be minimized. Numbers, including integers, should be floats with 32-bit (or less) precision, otherwise precision is lost during conversion to 32-bit.
- *poly_indices* - list of lists containing the indices for coefficient values for polynomial to be minimized.
- *num_variables* - optional input to specify number of variables for polynomial. Must be greater than or equal to maximum index value in *poly_indices*.
- *relaxation_schedule* - four different schedules represented in integer parameter. Higher values reduce the variation in the analog spin values and therefore, are more probable to lead to improved objective function energy for input problem. Accepts range of values in set {1, 2, 3, 4}.
- *sum_constraint* - a normalization constraint that is applied to the problem space that is used to calculate energy. This parameter will be rounded if exceeds float32 precision (e.g. 7-decimal places). Value must be between 1 and 10000.

For additional information see source documentation for function here. Note that the default port in the device and software is 50051.

Copyright 2026 Quantum Computing Inc. 

```
#!/usr/bin/env python

from eqc_direct.client import EqcClient

# MUST FILL IN THE VALUES IN THIS FROM YOUR NETWORKING SETUP FOR THE DEVICE
eqc_client = EqcClient(ip_address="localhost", port="50051")
poly_coefs = [-10, -10, -3.7, -5]
poly_indices = [[0,1], [0,3], [2,2], [1,3]]
# use lock to prevent other users from taking exclusive access to the device
lock_id, start_ts, end_ts=eqc_client.wait_for_lock()
try:
    result_dict = eqc_client.solve_sum_constrained(
        lock_id=lock_id,
        poly_indices = poly_indices,
        poly_coefficients = poly_coefs,
        relaxation_schedule = 2,
        num_samples = 2,
        sum_constraint = 100)
finally:
    # release lock when finished using the device
    lock_release_out = eqc_client.release_lock(lock_id=lock_id)
```

Each response as obtained in *result_dict* above will include the following fields:

- *err_code* - 0 if problem was solved without any issues, otherwise an integer greater than 0 representing the error that occurred
- *err_desc* - a short description of the error code for the submission
- *preprocessing_time* - time in seconds to validate data and re-format input data for running on the device.
- *runtime* - the time in seconds which the device required to solve the problem
- *postprocessing_time* - runtime for auxiliary computations that occur besides sampling during each sample routine from Dirac hardware including intermediate energy calculations, objective function adjustments, and distillation of solutions.
- *energy* - list of energies for best solution found (float32 precision) for each sample from Dirac hardware after error correction
- *solution* - a list of vectors representing the lowest energy solution (float32 precision) for each sample from Dirac hardware after error correction
- *num_variables* - specifies the number of variables in the polynomial in solved polynomial.
- *num_samples* - the number of samples (independent solutions) to generated from the device

Copyright 2026 Quantum Computing Inc.

- *calibration_time* - calibration time is unrelated to execution of the individual sampling for the optimization. This time is from system level interruptions from calibrations that happen at regular intervals to maintain system performance
- *start_job_ts* - nanosecond timestamp marking start of submission to device
- *end_job_ts* - nanosecond timestamp marking time at which results were acquired from device and returned to the user
- *raw_energy* - list of energies for best solution found (float32 precision) for each sample from Dirac hardware before error correction
- *raw_solution* - a list of vectors representing the lowest energy solution (float32 precision) for each sample from Dirac hardware before error correction

An example response for the problem above would appear as follows:


```
{'calibration_time': 0.0,
 'end_job_ts': 1789751066801436375,
 'energy': [-60.0, -60.0],
 'err_code': 0,
 'err_desc': 'Success',
 'num_samples': 2,
 'num_variables': 3,
 'postprocessing_time': [1.4899999314366141e-06, 1.4999999393694452e-06],
 'preprocessing_time': 0.004330518655478954,
 'raw_energy': [-59.9999962, -59.9999886],
 'raw_solution': [[1.9991271, 0.0, 2.0008729], [2.0019295, 0.0, 1.9980708]],
 'runtime': [0.7227075695991516, 0.7275634407997131],
 'solution': [[2.0, 0.0, 2.0], [2.0000002, 0.0, 2.0]],
 'start_job_ts': 1789751064747709709}
```


To run a problem using the integer solver see overview here, function spec for input function *solve_integer* here, and description of function output here.

Copyright 2026 Quantum Computing Inc.

# 3 Understanding Device Status

## 3.1 System Status

The system status provides information regarding whether or not the system is functioning as expected and also provides information on what operations the system is currently performing.

| System Status Name | System Status Code | Description |
|---|---|---|
| IDLE | 0 | device is not in use and is available for job submission |
| JOB_RUNNING | 1 | device is in computing mode |
| CALIBRATION | 2 | device is running a calibration routine. This process occurs only when the device is in an idle state and will temporarily make the system unavailable for problem submissions |
| HEALTH_CHECK | 3 | device is running health tests to assess the current hardware performance as well as detect potential issues with the hardware. This process occurs only when the device is in an idle state and will temporarily make the system unavailable for problem submissions |
| HARDWARE_FAILURE | 4, 5 | Device has entered a failed state and is no longer available to solve problems until hardware issues have been remediated |

**Table 1:** System status information summary.

Note that health checks and calibrations should not take more than a few minutes.

Copyright 2026 Quantum Computing Inc.

## 3.2 Submission/Results Statuses

Every single submission and results object contains fields *err_code* and *err_desc*. A nonzero error code indicates that the submission or result experienced that specific error while processing. If a nonzero error code is present the submission will not return any data from the Dirac-3 device.

| Error | Code | Description |
|---|---|---|
| `normal_code` | 0 | Success |
| `index_out_of_range` | 1 | Index in submitted data is out of range for specified number of variables. |
| `coef_index_mismatch` | 2 | Polynomial indices do not match required length for specified coefficient length. |
| `device_busy` | 3 | Device is currently processing another request. |
| `lock_mismatch` | 4 | `lock_id` does not match the current device lock. |
| `hardware_failure` | 5 | Device failed during execution. |
| `invalid_sum_constraint` | 6 | Sum constraint must be ≥ 1 and ≤ 10000. |
| `invalid_relaxation_schedule` | 7 | Parameter `relaxation_schedule` must be in the set {1, 2, 3, 4}. |
| `user_interrupt` | 8 | User sent stop signal before result was returned. |
| `exceeds_max_size` | 9 | Exceeds maximum problem size for device. |
| `decreasing_index` | 10 | One or more polynomial indices are not in non-decreasing order. |
| `invalid_precision` | 11 | Input precision exceeds the device's maximum allowed precision. |
| `num_samples_positive` | 12 | Input `num_samples` must be positive. |
| `precision_constraint_mismatch` | 13 | Sum constraint must be divisible by solution precision. |
| `precision_nonnegative` | 14 | Input solution precision cannot be negative. |
| `num_variables_positive` | 15 | Input `num_variables` must be greater than 0. |
| `degree_positive` | 16 | Input degree must be greater than 0. |
| `num_levels_num_vars_mismatch` | 17 | Length of `num_levels` must match `num_variables`. |

**Table 2:** Problem submission and results error information summary.

Copyright 2026 Quantum Computing Inc.

| Error | Code | Description |
|---|---|---|
| num_levels_gt_one | 18 | All elements of input 'num_levels' must be greater than 1 |
| total_integer_levels | 19 | Total number of integer levels from input variables num_levels exceed limit |
| invalid_mean_photon_number | 20 | Mean photon number if specified must be in range [0.0000667, 0.0066666] |
| invalid_quantum_fluctuation_coefficient | 21 | Quantum fluctuation coefficient if specified must be in range [1, 100] |

**Table 3:** Problem submission and results error information summary (continued).

# 4 Remote Command Line Interface for Administrators

To facilitate routine administrative tasks and enhance device security, QCi allows administrators to connect to the device via SSH. The designated user for administrative tasks is qciadmin, which has access to a custom Command Line Interface (CLI). This interface allows users to modify the device's networking configuration, change passwords, power down, or reboot the device. To access the device, first obtain its current IP address. Before it is reassigned the device is located 192.168.42.13/24. To cancel any input to a function in the CLI, simply press CTRL+C in the terminal. This action will interrupt and cancel the current command before it executes.

## 4.1 Remote login

Using the IP address you've identified for your device you will be able to SSH to the admin portal from the command line remotely using the following command, which will prompt you for a password which is on a sticker on the back panel of the device:

```
ssh qciadmin@YOUR_DEVICE_IP_ADDRESS
```

After logging in, the user will be presented with a list of available commands to execute. A list of all available commands can be found in the table below:

## 4.2 Changing password

It is recommended as a first step to change your password from the default value. To do this, input the number 4 in the terminal and press Enter. To reset your password, you will need to confirm your current password and then enter your new password twice for validation:

```
Enter your choice (0-9): 4
Changing password for qciadmin.
Current password:
New password:
Retype new password:
```

Copyright 2026 Quantum Computing Inc.

| Input # | Command | Action |
|---|---|---|
| 0 | Exit | Immediately ends user session and disconnects user from uQRNG CLI |
| 1 | Power off system | Shutdown system remotely however does not turn off power to the photonic portion must press button back panel to power down |
| 2 | Reboot system | Restarts system |
| 3 | Show current IP address | Prints current IP address to terminal |
| 4 | Change password | Allows user to change qciadmin password |
| 5 | Configure network interface | Allows user to modify network interface including: IP address, subnet, gateway, and DNS server |
| 6 | Show network interface config | Prints content of ethernet network file to terminal |
| 7 | Add a public key for SSH auth | Allows user to add a public key for authentication to the device for SSH |
| 8 | Delete a public key for SSH auth | Allows user to delete a public key that was available for authentication to the device for SSH |
| 9 | Show all public keys for SSH | Prints all public keys that are currently available for authentication |

**Table 4:** List of all available commands through admin terminal.

## 4.3 Changing the network configuration

**Warning!!!** Changing network configurations can lead to connectivity issues and may render your device unreachable if improperly configured (Notably, with uncareful use of DHCP). Improper network configuration may also expose your device to unauthorized access by external users.

Admin users are allowed to configure the networking of their device using the following attributes:

- **mode** - the operating mode for the Ethernet interface to the Dirac-3 device can be either static or DHCP
- Additional static configurations:
  - **IP address** - Pv4 formatted IP address
  - **subnet mask** - (CIDR format) 32-bit number used in IP networking to divide an IP address into network and host portions
  - **gateway** - specify device or a node on a network that serves as an access point to another network. It's optional, but if provided, it allows the device to communicate beyond

Copyright 2026 Quantum Computing Inc.

its local subnet. Must be IPv4 formatted.

- **dns server** - DNS servers within local networks (e.g., corporate networks, home networks) resolve internal domain names that are not publicly accessible on the internet. This capability facilitates efficient communication among devices within the same network. Currently only one can be specified for this input.

An example of using the CLI when specifying static can be seen below:

```
Enter your choice (0-9): 5
Enter the mode (static/dhcp): static
Enter IP address (IPv4): 192.168.42.42
Enter subnet mask (CIDR):
/24
Enter gateway (IPv4, optional, press Enter to skip):
Enter DNS servers (IPv4, optional, press Enter to skip):
```

# 5 Related Technical References

The following documents provide additional technical details that complement this User Guide:

- **Dirac-3S User Guide** Comprehensive guide to understand Dirac-3S computation, timing, how to install the hardware, and how to use the hardware.
- **Dirac-3S Benchmarking of Continuous Solver** Benchmark examples, test methodologies, performance metrics, and results.

# 6 Contacting Support

For assistance, please contact:
Email: support@quantumcomputinginc.com
Support Portal: https://quantumcomputinginc.ladesk.com/
When contacting support, provide:

- Dirac-3S serial number
- Software version
- Problem description
- Error messages

Copyright 2026 Quantum Computing Inc.

- Relevant log files

Copyright 2026 Quantum Computing Inc.