

Intel® Management Engine

Intel® Quiet System Technology Tools User's Guide

Revision 0.91

April 2008

Intel Confidential





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

Intel products are not intended for use in medical, life saving, or life sustaining applications.

Intel may make changes to specifications and product 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 Intel® Eaglelake chipset may contain design defects or errors, known as errata, which may cause the product to deviate from published specifications. Current characterized rrata are available on request.

Intel and the Intel logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

*Other names and brands may be claimed as the property of others.

Copyright © 2007 Intel Corporation. All Rights Reserved.



Contents

1 Intel Software License Agreement 8

2 Introduction 12

2.1 About This User's Guide 12
2.2 Terminology 12

3 Intel® Quiet System Technology Tools and Utilities 14

3.1 Configuration Tool (QstCfgD.exe and QstCfg.exe) 16
3.1.1 Usage 16
3.1.2 Implementation 16
3.2 Configuration GUI (QSTCT_GUI.exe) 18
3.2.1 Start Up - License Agreement 18
3.2.2 Main Program 19
3.2.3 Accessing Entities within an Entity Type 19
3.2.4 Editing Parameters 20
3.2.5 Help Area 21
3.2.6 Menu Bar 21
3.3 Weighting Matrix Utility 25
3.3.1 QstLog Tool (QstLog.exe) 25
3.3.1.1 Invocation 25
3.3.1.2 Operation 26
3.3.2 QstCtrl Tool (QstCtrl.exe) 26
3.3.2.1 Fan Speed Control 26
3.3.2.2 Data Display 26
3.3.2.3 Commands 27
3.3.3 MAXPower 27
3.3.4 Weighting Matrix Calculator (WeightingCalculationTool.exe) 28
3.3.4.1 Entering Fan-Sensor Configuration 28
3.3.4.2 Entering Collected Data 28
3.4 QstINI Tool (QstINID.exe and QSTINI.exe) 29
3.4.1 Invocation 29
3.5 Intel® QST Compliancy Tool (QstCply.exe) 30
3.5.1 Invocation 30
3.6 Header Files and Library Files 31
3.7 INI Files 31
3.8 Intel® QST Configuration Creation Wizard 32
3.8.1 Invocation 32
3.9 Tuning Wizard 40
3.9.1 Installation 40
3.9.1.1 Installing the Intel® MaxPower Program 40
3.9.1.2 Installing the Microsoft* .NET Framework 40
3.9.1.3 Installing the Intel® QST Tuning Wizard 41
3.9.2 Using the Configuration Wizard 41
3.9.2.1 Overview 41
3.9.2.2 Intel® QST Tuning Wizard Usage Process 41



3.9.2.3 PID Gain Setting Tuning 46
3.9.2.3.1 Determining the Quality of PID Gain Settings 46
3.9.2.3.2 Assessing Response Quality 46
3.9.2.3.3 Process for Adjusting the PID Gain Settings 49
3.9.2.4 Using the Plot Window 51
3.9.2.4.1 Plot Window Contents 51
3.9.2.4.2 Plot Window Options 52





Revision History

Revision Number	Description	Revision Date
0.8	Initial release	November 2007
0.81	Corrected Doc Title to “Intel® Management Engine”	January 2008
0.9	- Added Section 3.8 containing the user guides for the Configuration Creation Wizard. - Added Section 3.9 containing user guides for the Tuning Wizard.	March 2008
0.91	- Updated Section 3.7 to include all released default INI files. - Updated Section 3.9 to correct the installation file of the Tuning Wizard.	April 2008

§





1 Intel Software License Agreement

IMPORTANT - READ BEFORE COPYING, INSTALLING OR USING.

Do not use or load this software and any associated materials (collectively, the "Software") until you have carefully read the following terms and conditions. By loading or using the Software, you agree to the terms of this Agreement. If you do not wish to so agree, do not install or use the Software.

LICENSE. This Software is licensed for use only in conjunction with Intel component products. Use of the Software in conjunction with non-Intel component products is not licensed hereunder.

Subject to the terms of this agreement, Intel grants to you a nonexclusive, nontransferable, worldwide, fully paid-up license under Intel's copyrights to:

a) use, modify and copy Software internally for your own development and maintenance purposes; and

b) modify, copy and distribute Software, including derivative works of the Software, to your end-users, but only under a license agreement with terms at least as restrictive as those contained in Intel's Final, Single User License Agreement, attached as Exhibit A; and

c) modify, copy and distribute the end-user documentation which may accompany the Software, but only in association with the Software.

If you are not the final manufacturer or vendor of a computer system or software program incorporating the Software, then you may transfer a copy of the Software, including derivative works of the Software (and related end-user documentation) to your recipient for use in accordance with the terms of this agreement, provided such recipient agrees to be fully bound by the terms hereof. You shall not otherwise assign, sublicense, lease, or in any other way transfer or disclose Software to any third party. You shall not reverse-compile, disassemble or otherwise reverse-engineer the Software.

Except as expressly stated in this agreement, no license or right is granted to you directly or by implication, inducement, estoppel or otherwise. Intel shall have the right to inspect or have an independent auditor inspect your relevant records to verify Your compliance with the terms and conditions of this agreement.

CONFIDENTIALITY. If you wish to have a third party consultant or subcontractor ("Contractor") perform work on your behalf which involves access to or use of Software, You shall obtain a written confidentiality agreement from the Contractor



which contains terms and obligations with respect to access to or use of Software no less restrictive than those set forth in this agreement and excluding any distribution rights, and use for any other purpose. Otherwise, you shall not disclose the terms or existence of this agreement or use Intel's name in any publications, advertisements, or other announcements without Intel's prior written consent. You do not have any rights to use any Intel trademarks or logos.

OWNERSHIP OF SOFTWARE AND COPYRIGHTS. Title to all copies of the Software remains with Intel or its suppliers. The Software is copyrighted and protected by the laws of the United States and other countries, and international treaty provisions. You may not remove any copyright notices from the Software. Intel may make changes to the Software, or to items referenced therein, at any time and without notice, but is not obligated to support or update the Software. Except as otherwise expressly provided, Intel grants no express or implied right under Intel patents, copyrights, trademarks, or other intellectual property rights. You may transfer the Software only if the recipient agrees to be fully bound by these terms and if you retain no copies of the Software.

LIMITED MEDIA WARRANTY. If the Software has been delivered by Intel on physical media, Intel warrants the media to be free from material physical defects for a period of ninety (90) days after delivery by Intel. If such a defect is found,return the media to Intel for replacement or alternate delivery of the Software as Intel may select.

EXCLUSION OF OTHER WARRANTIES. EXCEPT AS PROVIDED ABOVE, THE SOFTWARE IS PROVIDED "AS IS" WITHOUT ANY EXPRESS OR IMPLIED WARRANTY OF ANY KIND INCLUDING WARRANTIES OF MERCHANTABILITY, NONINFRINGEMENT, OR FITNESS FOR A PARTICULAR PURPOSE. Intel does not warrant or assume responsibility for the accuracy or completeness of any information, text, graphics, links or other items contained within the Software.

LIMITATION OF LIABILITY. IN NO EVENT SHALL INTEL OR ITS SUPPLIERS BE LIABLE FOR ANY DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, LOST PROFITS, BUSINESS INTERRUPTION, OR LOST INFORMATION) ARISING OUT OF THE USE OF OR INABILITY TO USE THE SOFTWARE, EVEN IF INTEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. SOME JURISDICTIONS PROHIBIT EXCLUSION OR LIMITATION OF LIABILITY FOR IMPLIED WARRANTIES OR CONSEQUENTIAL OR INCIDENTAL DAMAGES, SO THE ABOVE LIMITATION MAY NOT APPLY TO YOU. YOU MAY ALSO HAVE OTHER LEGAL RIGHTS THAT VARY FROM JURISDICTION TO JURISDICTION.

TERMINATION OF THIS AGREEMENT. Intel may terminate this Agreement at any time if you violate its terms. Upon termination, you will immediately destroy the Software or return all copies of the Software to Intel.

APPLICABLE LAWS. Claims arising under this Agreement shall be governed by the laws of California, excluding its principles of conflict of laws and the United Nations Convention on Contracts for the Sale of Goods. You may not export the Software in violation of applicable export laws and regulations. Intel is not obligated under any other agreements unless they are in writing and signed by an authorized representative of Intel.

GOVERNMENT RESTRICTED RIGHTS. The Software is provided with

"RESTRICTED RIGHTS." Use, duplication, or disclosure by the Government is subject to restrictions as set forth in FAR52.227-14 and DFAR252.227-7013 et seq. or their successors. Use of the Software by the Government constitutes acknowledgment of

Intel's proprietary rights therein. Contractor or Manufacturer is Intel Corporation, 2200 Mission College Blvd., Santa Clara, CA 95052.

§



2 Introduction

2.1 About This User's Guide

This document is intended for Original Equipment Manufacturers (OEM) and System Integrators. The document introduces and provides details on how to use different tools for the Intel® Quiet System Technology (Intel® QST). The tools are included in different management engine firmware kits provided by Intel. These tools are provided to facilitate the configuring and fine tuning the Intel® QST subsystem. For more information on the Intel® QST configuration and fine tuning, refer to the latest version of the Intel® Quiet System Technology Configuration and Tuning Manual.

2.2 Terminology

Term	Description
Intel® QST	Intel® Quiet System Technology
ATX	Advanced Technology eXtended form factor.
BTX	Balanced Technology eXtended form factor.
Firmware (FW)	Embedded firmware running on the Intel® Management Engine
Graphic User Interface (GUI)	A visual framework that allows the user to interact with an application through direct manipulation of graphics objects.
Intel® Management Engine (ME)	The embedded processor located in the (G)MCH

§



3 Intel® Quiet System Technology Tools and Utilities

Intel provides a collection of tools and utilities to enhance the ease-of-use of the configuration and tuning process for the Intel® QST subsystem. These tools and utilities include:

1. **Configuration Tool (QstCfgD.exe and QstCfg.exe) and GUI (QSTCT_GUI.exe)**: used by OEM/ODM and system integrators to configure and fine tune the Intel® QST subsystem.

2. **Weighting Matrix Utility**: used for fine tuning purposes to determine the best way to map sensors to fans on specific platforms for optimum acoustic performance. The utility is a collection of individual tools working together, including:

   a. Processor MAXPower application: used to generate specific power loads on the processor and system.

   b. Intel® QST Fan Control (QstCtrl.exe) tool: used to manually control fan speed.

   c. Intel® QST Logging (QstLog.exe) tool: used to periodically collect current readings from the Temperature, Voltage and Fan Speed Sensors and duty cycle values from the Fan Speed Controllers.

   d. Weighting Matrix Calculator (WeightingCalculationTool.exe): an application that calculates the fan weighting values based upon the data collected on the system.

3. **Configuration File Reconstitution tool (QstINI.exe)**: used to verify configurations stored in binary files and within the Intel® QST Subsystem.

4. **Intel® QST Compliancy Tool (QstCply.exe)**: used to check the readiness of Intel® QST subsystem.

5. **Header files and library files**: these components facilitate the development of new host applications to communicate with the Intel® QST firmware in monitoring health data from the Intel® QST subsystem.

6. **ATX and BTX INI files (QstCfgATX.ini and QstCfgBTX.ini)**: these are the initial INI files that facilitate the Intel® QST configuration process.

7. **System Tools**: these tools facilitate the Intel® QST configuration process and include:

   a. Flash Image Tool (Ftoolc.exe): used to combine individual images, such as BIOS image, Intel® QST firmware, Intel® QST payload, into one single image for storing onto SPI flash.



b. Flash Programming Tool (Fprog.exe): used to write an image to SPI flash or update an image in the SPI flash device.

NOTE: This document does not cover how to use the system tools. For details on how to use the system tools (Ftoolc and Fprog), refer to the System Tools User's Guide included in the management engine firmware kit.



3.1 Configuration Tool (QstCfgD.exe and QstCfg.exe)

QstCfgD.exe: Pure DOS command line tool. The tool is supported on Free DOS and Windows 98. This tool requires DOS4GW, which can be downloaded from the following URL:
http://www.scene.org/file.php?file=%2Fresources%2Fdos4gw.exe&fileinfo

QstCfg.exe: Windows command line tool. The tool is supported on Windows 2000 SP4, Windows XP SP1, Windows XP SP2, and Windows XP 64.

3.1.1 Usage

The Configuration Tool can be utilized to achieve different goals, including:

1) Initially configuring Intel® QST subsystem: The Configuration Tool can generate a binary payload from an INI file containing initial control parameters. The binary payload can then be programmed into the SPI flash to configure Intel® QST subsystem.

2) Tuning Intel® QST subsystem: after initially configuring the Intel® QST subsystem, the parameters in the INI file can be modified and optimized, based upon specific system design and thermal characteristics, to achieve optimum performance of the Intel® QST Subsystem. The Configuration Tool can be used to deliver INI files with optimized settings to the ME for subsequent storage in the SPI flash. The Configuration Tool can be easily integrated into the factory floor tool/application to automatically store the final settings into the SPI flash of production systems in manufacturing line processes.

3) Intel® QST testing and validation: during testing, validation and tuning of the Intel® QST subsystem, control parameters may need to be changed often, in order to optimize Intel® QST performance. The Configuration Tool provides an easy way to update the settings of the Subsystem.

3.1.2 Implementation

The primary environments for the execution of the Configuration Tool is the Windows* O/S and DOS. The Configuration Tool takes an INI file containing configuration settings and can generate a binary payload file to be subsequently programmed into the SPI flash or directly store a generated binary payload into the SPI flash, using the services of the Management Engine (ME).

The Configuration Tool will execute from the command line. Its operation is defined as follows:

1. The Tool will parse the configuration contained within the specified INI file and verify its contents. If an error is detected, a report will be generated and the program's execution terminated.



2. The Tool will direct its output (status updates, error messages, etc.) to STDOUT. If the command line redirects the program's output to a (log) file, the program will also direct a copy of its output to STDERR, so that it is always visible to the operator.

3. If the Tool successfully parses the configuration, it will prepare a binary payload for subsequent delivery to the Intel® QST Subsystem. Support will be provided for this payload to be delivered directly to the ME for processing or for it to be written to a file (for direct storage into flash, along with the ME's F/W).

The syntax for the tools is defined as follows:

QstCfgD <INI File Pathname> [{-w|-o} <Binary File Pathname>]
(Ex: QstCfgD QstCfgATX.ini –w QSTBinary.bin)

QstCfg <INI File Pathname> [{-w|-o} <Binary File Pathname>]
(Ex: QstCfg QstCfgATX.ini –w QSTBinary.bin)

Where:

INI File Pathname	Specifies the pathname for the INI file.
-w <Binary File Pathname>	Specifies that the payload is to be written to a file. If the file already exists, the tool will ask for confirmation before overwriting it.
-o <Binary File Pathname>	Specifies that the payload is to be written to a file. If the file already exists, it will be overwritten.

Notes:

- If the optional INI File Pathname is not specified, the default pathname will be "**Qstcfg.ini**".
- If the (optional) Binary File Pathname is specified, the payload will be written to this file. If no pathname is specified, the default pathname will be "**Qstcfg.bin**".
- Parameters "-w" and "-o" are mutually exclusive. Only one of them may be specified.
- If neither the "-w" or "-o" parameters are specified, the binary payload will be delivered to the ME for processing. This will be a synchronous operation. That is, the program will wait for and report whether the payload was accepted or rejected by the Intel® QST Subsystem.

Windows Command Line Examples:

1. **"qstcfg"**: Specifies that the configuration contained within file "Qstcfg.ini" will be parsed and, if no errors are detected, delivered to the ME for processing.

2. **"qstcfg config.ini"**: Specifies that the configuration contained within file "config.ini" will be parsed and, if no errors are detected, delivered to the ME for processing.

3. **"qstcfg –o"**: Specifies that the configuration contained within file "qstcfg.ini" will be parsed and, if no errors are detected, written to file "qstcfg.bin". If file "qstcfg.bin" already exists, it will be overwritten.

4. **"qstcfg config.ini –w config.bin"**: Specifies that the configuration contained within file "config.ini" will be parsed and, if no errors are detected, written to file "config.bin". If file "config.bin" already exists, permission will be sought for overwriting it.

3.2 Configuration GUI (QSTCT_GUI.exe)

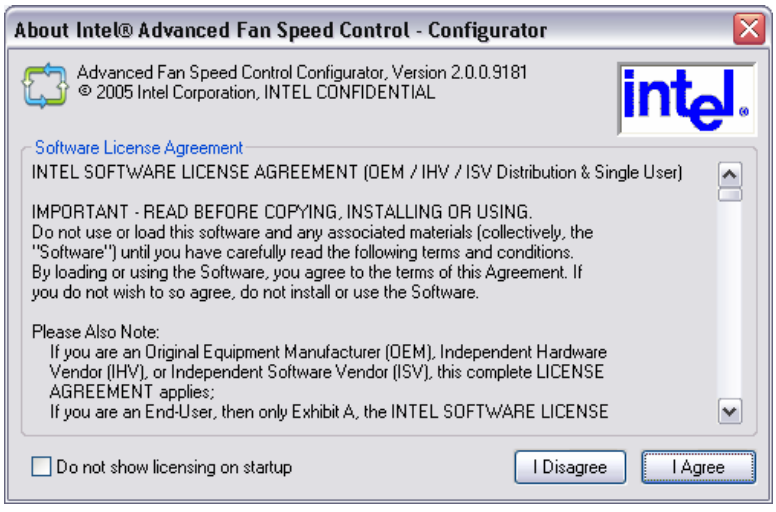
This is a Graphical User Interface (GUI) to the Intel® QST Configuration tool. The GUI is a Win32 executable file. It takes Intel® QST configuration parameters which can be manually entered or extracted from an INI file. Via this traversable structure, these configuration variables may be edited and saved. The tool can also generate an INI file from the variable values that is entered or edited through the GUI.

NOTE: for Window* 2000 SP4 OS, the tool requires Microsoft MSVCP60.dll. Make sure MSVCP60.dll is installed on the system.

The following sections describe in detail how to use the Configuration GUI.

3.2.1 Start Up - License Agreement

When the program is started, Intel's license agreement is shown in a scrollable frame:

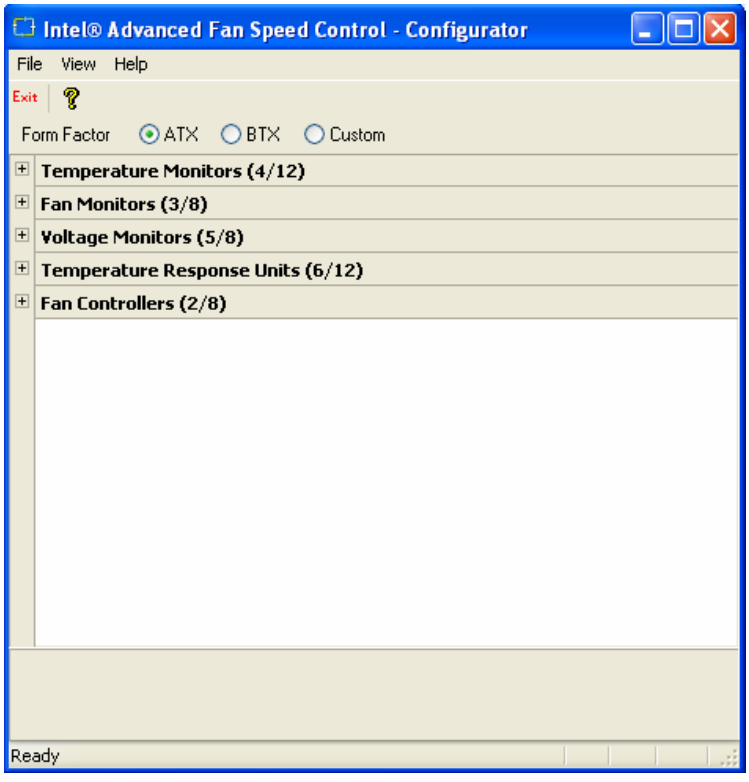


Clicking “I Agree” will open the main program. Clicking “I Disagree” causes the program to exit. Checking “Do not show licensing on startup” before clicking on “I Agree” will disable the display of the License Agreement during subsequent invocations of the program.

3.2.2 Main Program

The main frame of the GUI contains collapsible trees for each entity type. There are five entity types, categorized by functionality: Temperature Monitors, Fan Monitors, Voltage Monitors, Temperature Response Units, and Fan Controllers.

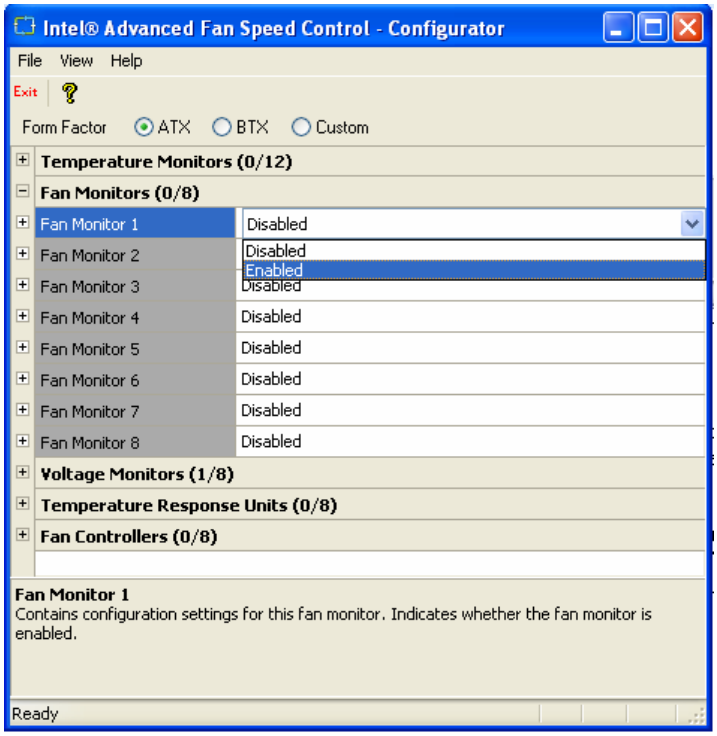
Under each entity type is a collection of entities. There is an indication of how many entities are supported and how many of them are enabled. For example, (4/12) after the Temperature Monitors entity type in the figure below indicates that there are a total of 12 temperature monitors and that 4 of them are currently enabled.



3.2.3 Accessing Entities within an Entity Type

Each entity type, when expanded by clicking on the [+] symbol, will show all the entities available of that type. Each entity can be enabled or disabled by clicking on the white bar next to it and choosing an appropriate option from the pull-down list.

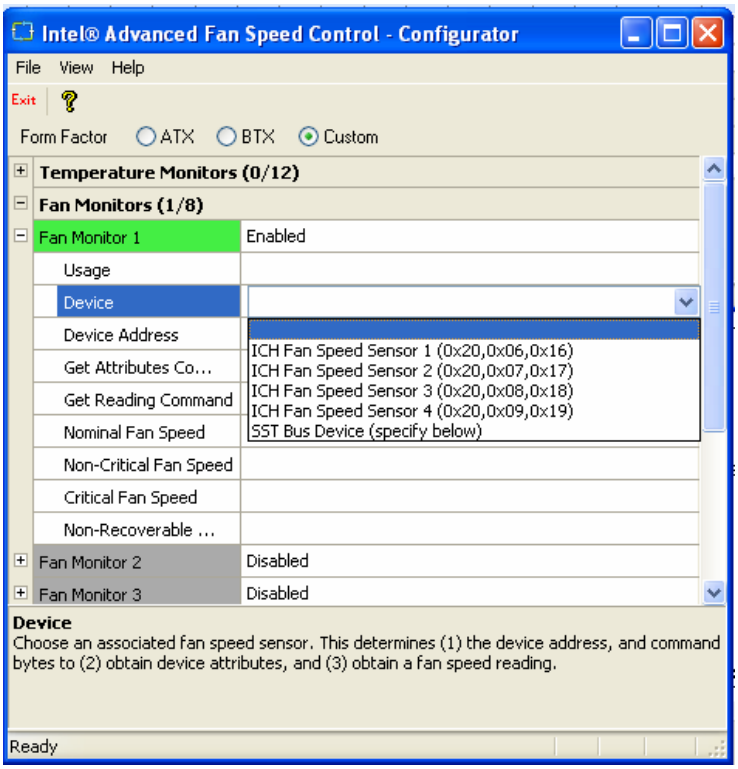
As illustrated in the figure below, after expanding the Fan Monitors type, 8 Fan Monitor entities can be accessed. Each entity can be enabled or disabled through the pull-down list.



3.2.4 Editing Parameters

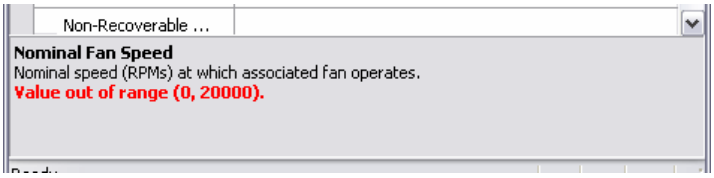
When an entity is enabled, the parameters for that entity can be accessed by clicking on the [+] symbol. To edit the parameter, left-click into an edit bar and either enter or choose a new value from the pull-down list.

Note that if an entity is disabled, the parameters for that entity cannot be edited.



3.2.5 Help Area

The help area at the bottom of the main window conveys information about the selected entity or parameter. Information is provided in standard black text and status messages are provided in red.



3.2.6 Menu Bar



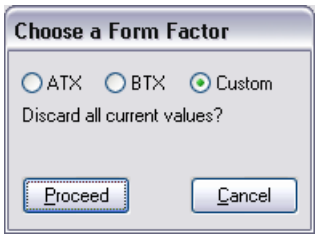
The menu bar of the GUI includes File, View, and Help.

This section covers each menu option. A brief description for each menu item can also be seen in the status bar as the mouse cursor hovers over it.

File → New

Values present in the various entities will be wiped (the user will be prompted to save these values if he has not done so) and replaced with a default set of values. Three types of default values exist (categorized by the form factor of the system being configured):

- **ATX:** Set of Intel recommended defaults for ATX
- **BTX:** Set of Intel recommended defaults for BTX
- **Custom:** All fields are blank

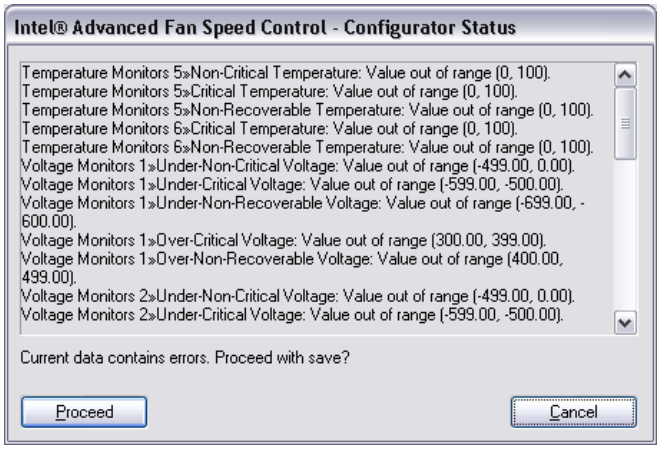


File → Load INI File…

Values present in the entity records will be wiped (the user will be prompted to save these values if he has not done so) and replaced with values from an INI file chosen by the user. A popup will ask which form factor is desired.

File → Save INI File…

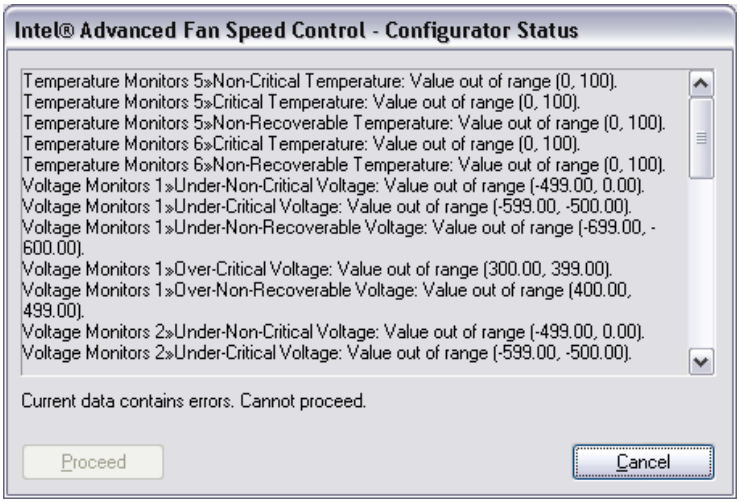
This option saves current values in the entity records to an INI file of the user's choice. A popup box will be displayed if the Configurator finds issues with any field values (see figure below). The user has the option to Cancel in order to make edits before saving, or to Proceed and save with errors.



If an existing INI file is chosen, parameters will only be saved to it if they are changed. Comments in an existing INI file are preserved unless they are on line along with a parameter that is changed.

File → Generate Configuration Payload…

This option saves current values in the fields to a binary payload file (.BIN) of the user's choice. A popup box will be displayed if the Configurator finds issues with any parameter values (see figure below). In this case, the user will have only the option to Cancel the Save operation, in order to make corrections before saving. The Configurator will only enable the option to proceed if the user's data is error-free.



File → Send Configuration to Firmware…

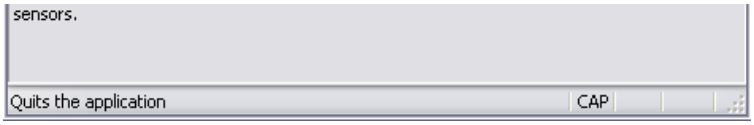
This option sends the parameters directly to the firmware. As with "File → Generate Configuration Payload", the user cannot deliver data to firmware if errors are found by the Configurator.

File → Exit

Choose this option to exit the program.

View → Status Bar

Toggles the visibility of the status bar at the bottom of the window



View → Toolbar

This option toggles the visibility of the toolbar right under the menu bar.

View → Expand All

This option expands all tree nodes so that every parameter is made directly available. The scrollbar at the right of the window can be used to access parameters that are not immediately visible.

View → Collapse All

This option collapses all tree nodes so that only the root Entity Types are directly available. Clicking on the **[+]** symbol that marks a collapsible tree will expand the tree.



3.3 Weighting Matrix Utility

The Weighting Matrix utility is a collection of tools that are provided by Intel. These tools include the QstLog, QstCtrl, MaxPower, and Weighting Matrix Calculator. The tools work together to determine the best way to map sensors to fans and to determine the weighting of each fan response. The Weighting Matrix utility can be used by the OEM/ODM during Intel® QST subsystem configuration or by system integrators during Intel® QST fine tuning process. The utility works for any system form factor and number of fans and sensors. **For information on how to use these tools to determine weighting matrix values during fine-tuning process, refer to the latest Intel® Quiet System Technology Configuration and Tuning Manual. Contact your Intel representative to get access to the document.**

The following sections detail the usages of these individual tools.

3.3.1 QstLog Tool (QstLog.exe)

This tool provides support for the periodic collection of current readings from the Temperature and Fan Speed Sensors and current duty cycle values from the Fan Speed Controllers. Optionally, voltage readings may be collected as well. The tool will sample the available Sensors and Controllers once per second and output the readings and settings both to the log file and to the display.

3.3.1.1 Invocation

The tool is an executable file working in Windows environment. It executes in from the command line. Its command line syntax is defined as follows:

QstLog [-L <Log-Path>] [-M] [-T <Samples>] [-V]

Where:

-L <Log-Path>	Specifies the pathname for the log file. This file will be overwritten. No warning of overwrite will be provided. Note: If no Pathname is provided, the default pathname will be "QstLog.txt"
-M	Specifies that the sample set collected in each one-second period is to be output to the console on a separate display line.
-T <Samples>	Specifies that the program is to terminate after logging the specified number of sampled reading sets.
-V	Specifies that voltage readings are to be included in the data displayed and logged. By default, only temperatures, fan speeds and fan duty cycles are collected, displayed and logged.

Example: "C:\Program Files\Qst\QstLog.exe" –L "C:\QstLog.txt"

This command will display the readings of temperatures, fan speeds and fan duty cycles of the available sensors and fan controllers. Also, the readings will be saved in file QstLog.txt in the root directory of drive C.

3.3.1.2 Operation

1. The tool will first verify that the Intel® QST Subsystem is configured. If it is not, operation cannot continue.
2. The tool will enumerate the supported sensors and controllers. Labels describing the usage for the sensors will be output to the log file (in the same order that readings/settings will be output). As well, terse labels will also be output to the console window.
3. Readings will be sampled from all available Sensors and Duty Cycle values will be sampled from all available Fan Speed Controllers. The readings will be output to the log file as well as to the console display. A sequence number is also included in the log file data.
4. Step 4 will be repeated, once per second, until a key press is detected or until the specified number of sample sets has been collected.
5. When a key is pressed, the log file will be closed. A summary, providing minimum, maximum and average readings and duty cycle values will be output to the console.
6. The tool will wait for the <enter> key to be pressed before terminating. This will provide the operator with an opportunity to peruse the summary information generated.

3.3.2 QstCtrl Tool (QstCtrl.exe)

This tool provides support for the manual specification of the settings (duty cycle values) of the Fan Speed Controllers. It also provides support for the display of configuration and reading/setting information for the Temperature, Voltage and Fan Speed Sensors and for the Fan Speed Controllers.

3.3.2.1 Fan Speed Control

- The tool will provide support for individually modifying the Duty Cycle settings of all available Fan Speed Controllers.

- The tool will accept Duty Cycle values with decimal places; since most Fan Speed Controllers support them (their typical implementation uses values in the range 0 to 255 to represent percentages in the range 0 to 100).

3.3.2.2 Data Display

- The tool will provide the ability to display information about the available Sensors and Fan Speed Controllers, including usage and current readings and duty cycle settings.

- The tool will provide the ability to separately or inclusively display temperature readings, fan speed readings, voltage readings and/or fan controller duty cycle settings.



3.3.2.3 Commands

The tool runs in interactive mode, providing a command prompt and allowing the operator to perform multiple operations. Operation will continue until the operator enters an ***exit*** command. QstCtrl will be initiated from the command line. It has no command-line parameters.

QstCtrl will support the following internal commands:

1. **a <ctrl>** — Requests that the specified Fan Speed Controller be placed in automatic control mode. The Intel® QST Subsystem will determine the Duty Cycle values to be utilized. **(Ex: a 1 – put Fan Speed Controller #1 in automatic mode.)**
2. **c** — Requests the display of information about all Fan Speed Controllers.
3. **e** — Requests that the tool terminate.
4. **f** — Requests the display of information about all Fan Speed Sensors.
5. **m <ctrl> <duty>** — Requests that the Duty Cycle output of the specified Fan Speed Controller be manually set to the specified value. For example, command “M 1 50” requests that Fan Controller 1 be operated at 50% duty cycle. **(Ex: m 1 50 – Fan speed controller #1 is manually set to 50% duty cycle.)**
6. **s** — Requests the display of information about all Sensors.
7. **t** — Requests the display of information about all Temperature Sensors.
8. **v** — Requests the display of information about all Voltage Sensors.

3.3.3 MAXPower

MAXPower Utility tool runs the processor at designated load.

There is a separate MAXPower utility (.exe execution file) for each processor. Make sure that the right version of the tool is installed for the target processor.

After installing the MAXPower program (double click on the .exe file), the desired load can be set by choosing: Options → Workload Configuration.

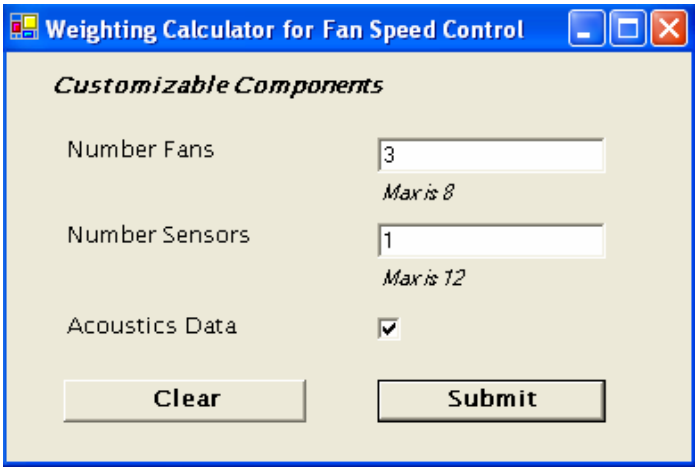
3.3.4 Weighting Matrix Calculator (WeightingCalculationTool.exe)

The WeightingCalculationTool is a Windows* executable file. It is used in the process of determine the weighting matrix values for a system. The calculator will calculate the appropriate weighting matrix values based on fan, sensor configuration and the data collected on a system.

3.3.4.1 Entering Fan-Sensor Configuration

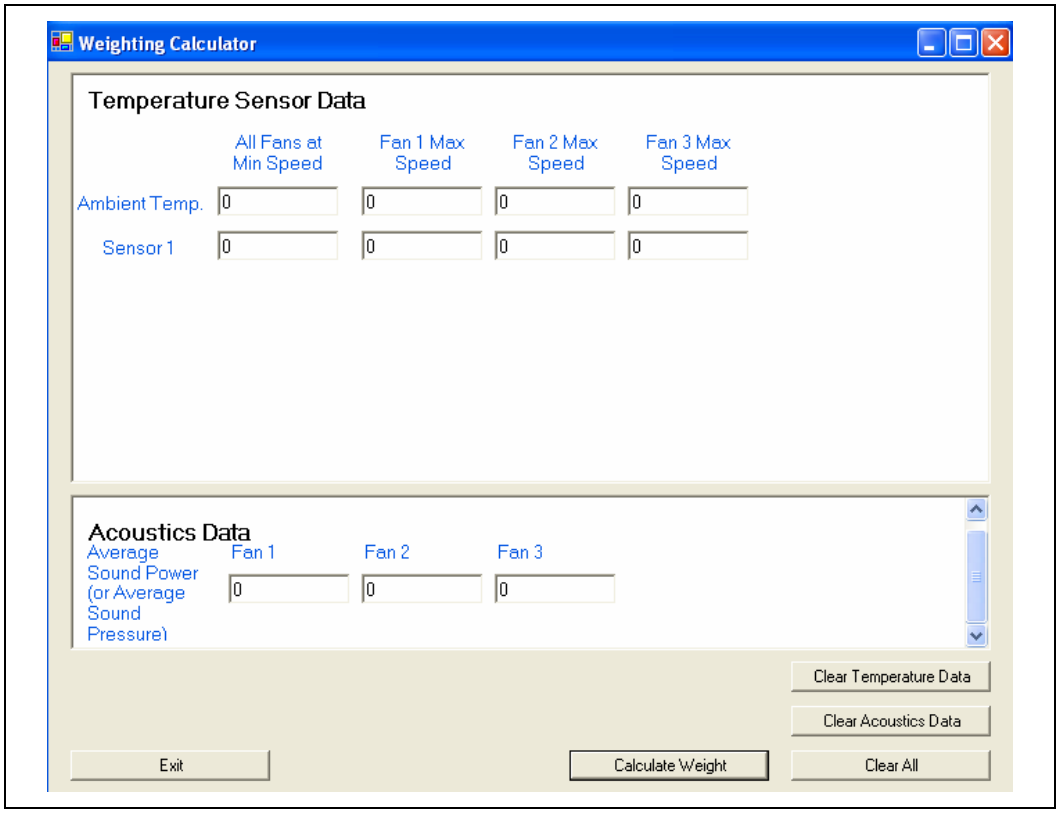
When the Weighting Calculator is opened, it will ask for fan and sensor configuration on the system. Enter the number of fans and sensors implemented on the system. If acoustics data is available, check the Acoustics Data box.

Hit <Submit> when done.



3.3.4.2 Entering Collected Data

After the fan-sensor configuration is submitted, a window will pop up with fields to enter data. Enter appropriate data collected from the platform. Hit the <Calculate Weight> button when finished. The weighting values calculated, based on the collected data, will be displayed.



3.4 QstINI Tool (QstINID.exe and QSTINI.exe)

This tool provides support for a level of post-mortem analysis of Intel® QST Subsystem configuration. It provides the ability to reconstitute an INI file from a binary configuration payload. This payload may come from a file created previously or may be directly obtained from the Intel® QST Subsystem. Once an INI file has been reconstituted, it can be compared to other INI files, in order to determine differences or, in the latter case, to verify that the configuration in use by the Intel® QST Subsystem is as expected.

QstINID.exe: Pure DOS command line tool. The tool is supported on Free DOS and Windows 98. This tool requires DOS4GW, which can be downloaded from the following URL:
http://www.scene.org/file.php?file=%2Fresources%2Fdos4gw.exe&fileinfo

QstINI.exe: Windows command line tool. The tool is supported on Windows 2000 SP4, Windows XP SP1, Windows XP SP2, and Windows XP 64.

3.4.1 Invocation

The tool is an executable file working in Windows environment. It executes in from the command line. Its command line syntax is defined as follows:

QstINID [-P <Payload-Path>] [-I <INI-Path>]
(Ex: QstINID –P QstBinary.bin)

QstINI [-P <Payload-Path>] [-I <INI-Path>]
(Ex: QstINI –I CurrentCfg.ini)

Where:

-P <Payload-Path>	Specifies the pathname for a Binary Payload File. If one is specified, the INI file will be reconstituted from its contents. If no path is specified, the current Configuration Payload will be obtained from the Intel® QST Subsystem and used to reconstitute the INI file.
-I <INI-Path>	Specifies the pathname for the resulting INI file. If no pathname is specified, default pathname "QstCfg.ini" will be used.

3.5 Intel® QST Compliancy Tool (QstCply.exe)

This tool checks the readiness of the QST-enabled platform by performing the following tests: General Chipset tests, Intel® Management Engine Interface Device tests, Chipset Thermal device tests, Intel® Management Engine Interface tests, Intel® Management Engine Firmware tests and Intel® QST tests.

At the end of the test, the tool will output PASS/FAIL results for each areas tested.

3.5.1 Invocation

The tool is executed in pure DOS and support Free DOS and Windows 98. This tool requires DOS4GW, which can be downloaded from the following URL:
http://www.scene.org/file.php?file=%2Fresources%2Fdos4gw.exe&fileinfo

Its command line syntax is defined as follows:

QstCply [/?] [/a] [/d] [/p] [/t]
(Ex : QstCply /a /t)

Where:

/?	Display help
/a	Dump Intel® QST Update Command Data
/d	Dump Intel® Management Engine Interface PCI Configuration Space.
/p	Dump all PCI Configuration Space
/t	Run Intel® QST Compliance test suite (Default)



3.6 Header Files and Library Files

These files are included in the 'QST Tools\Include' folder to facilitate the development of a host application to communicate with the Intel® QST firmware.

The header files available are:

1) QstCmd.h Defines data structures and associated literals for Intel® QST Subsystem
commands.

2) QstCfg.h Defines data structures and associated literals for Intel® QST Configuration
Payload.

3) QstInst.h : Defines data structures and associated literals, as well as providing function
prototypes, for the Instrumentation Layer (QstInst.dll).

4) QstComm.h Provides function prototypes for the Communications Layer (QstComm.dll).

5) typedef.h Defines standard data types used by the Intel® QST Subsystem.

The library files available are:

1) QstComm.libLibrary for static linking to QstComm.dll

2) QstInst.lib Library for static linking to QstInst.dll

3.7 INI Files

There are four initial INI files included in the kit:

1) **QSTCfgATXEL10.ini** for ICH10 Consumer based ATX form factor.

2) **QSTCfgATXEL10D.ini** for ICH10 Corporate based ATX form factor.

3) **QstCfgBTXEL10.ini** for ICH10 Consumer based BTX form factor.

2) **QstCfgBTXEL10D.ini** for ICH10 Corporate based BTX form factor.

These INI files are based upon the design of the Intel customer reference board. They can be used as the starting point of the configuration process as most of the control parameters can be used as they are, except for certain hardware dependent parameters. Those hardware-dependent parameters need to be modified by the designer to reflect specific hardware implementation on the platform. For example, device address of a sensor, how many fan headers implemented on the mother board, etc.

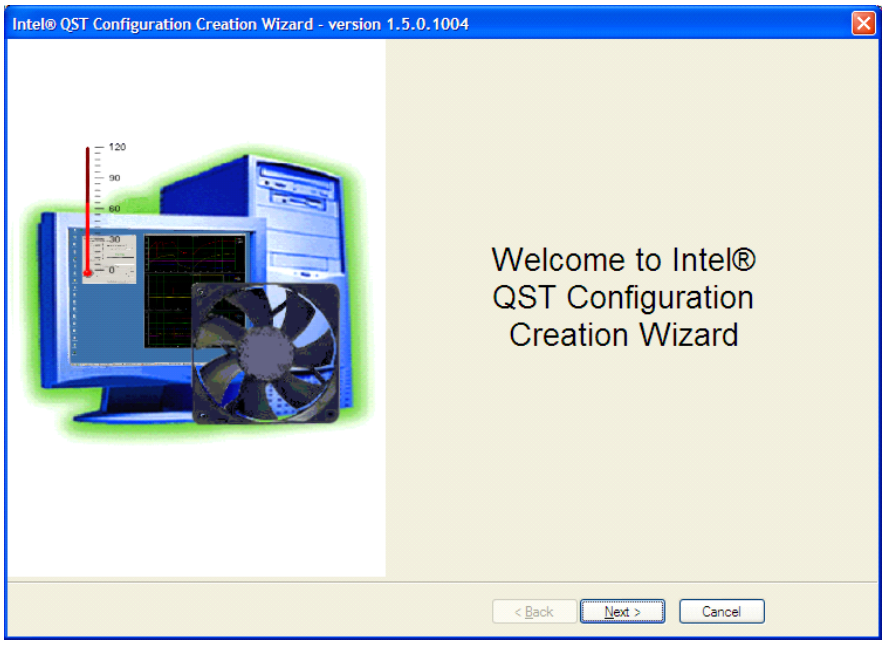
Note: the Configuration Creation Wizard, which is also included in the ME firmware kit, can be used to easily create an initial INI file for a specific platform without using one of these initial INI files and modifying needed parameters. Refer to section 3.8 for more details.

3.8 Intel® QST Configuration Creation Wizard

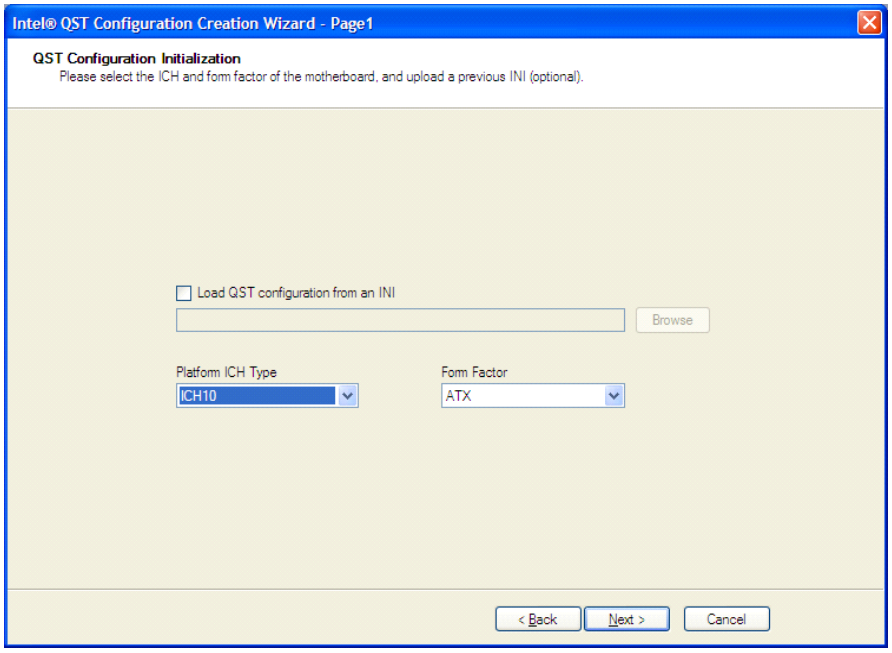
The Intel® QST Configuration Wizard is a tool that facilitates the configuration of the Intel® QST INI file. The tool aids the user in establishing the initial control settings and connecting sensor addresses to variables used by Intel® QST. The usage of this tool will primarily be by motherboard manufacturers or engineers responsible for configuring firmware within the motherboard. This tool does not provide system level tuning, which is facilitated by the Intel® Quiet System Technology (QST) Tuning Tool available from the ME Firmware Kit.

3.8.1 Invocation

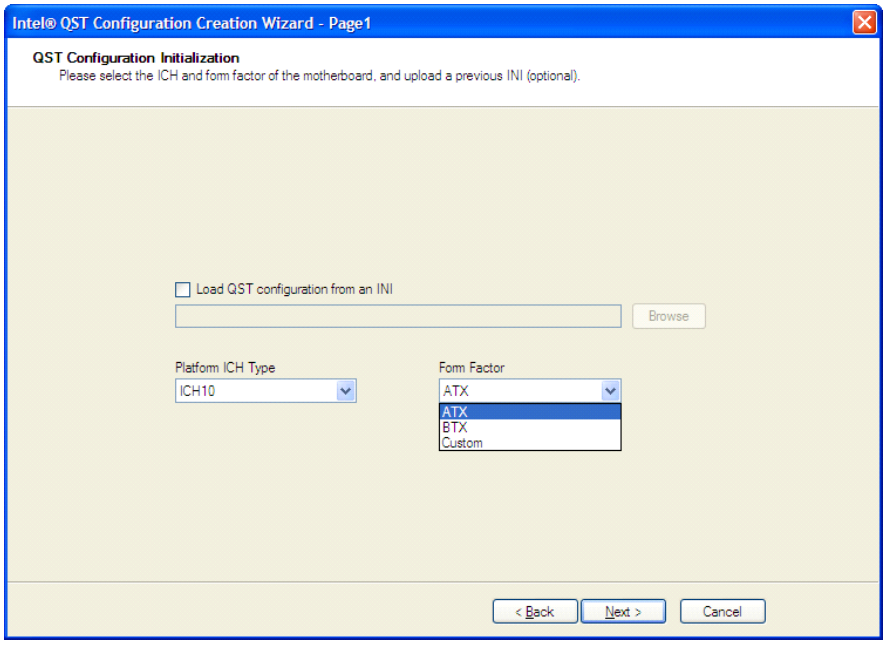
The tool generates the INI file containing control parameters for a specific platforms based upon inputs from the board designers. This INI file includes device addresses and commands used to read sensors and control fans. The following screen images walk the reader through the usage of the wizard.



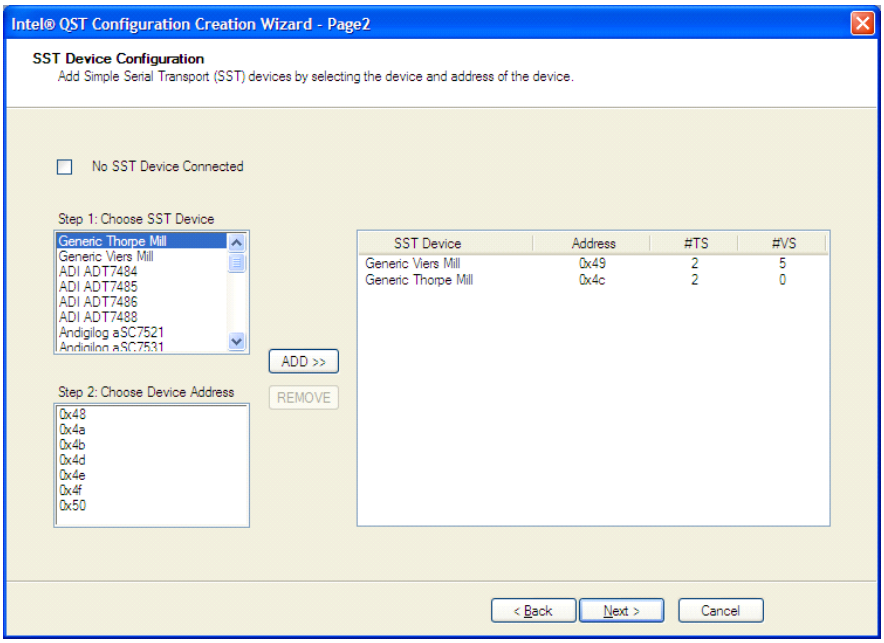
The Welcome screen will be the first screen the user will see once the Configuration Creation Wizard is started. The next screen prompts the user to select the ICH and form factor of the board through the drop down windows. Loading a previous INI configuration file will populate the current configuration with the various sensors and settings of the previous file.



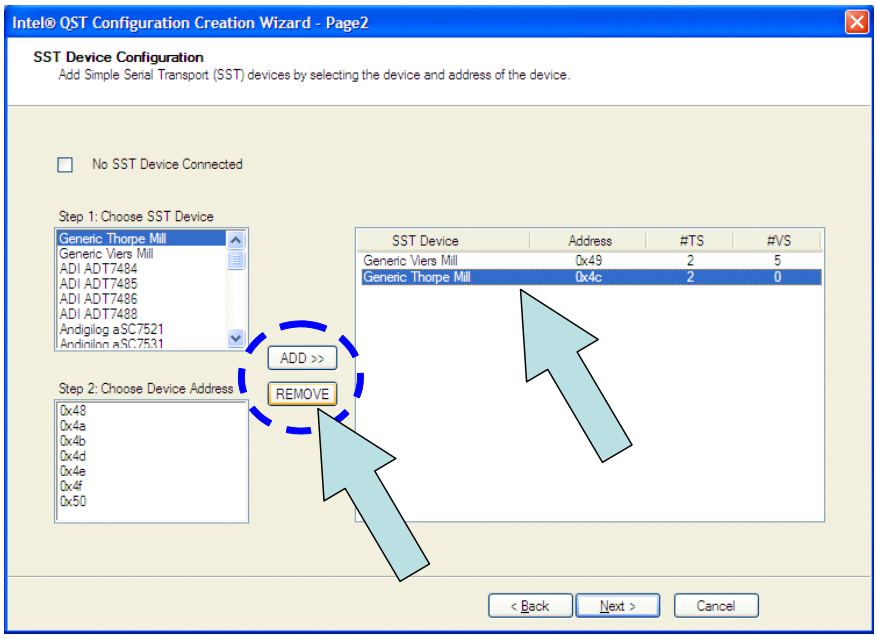
Two standard form factors, ATX and BTX, are available to select from the drop down shown in the next figure. These form factors encompasses all of the available sizes, such as micro-ATX. For nonstandard form factors, the “Custom” option would be selected. The form factor selection loads default settings that apply to the given form factor, such as PID settings.



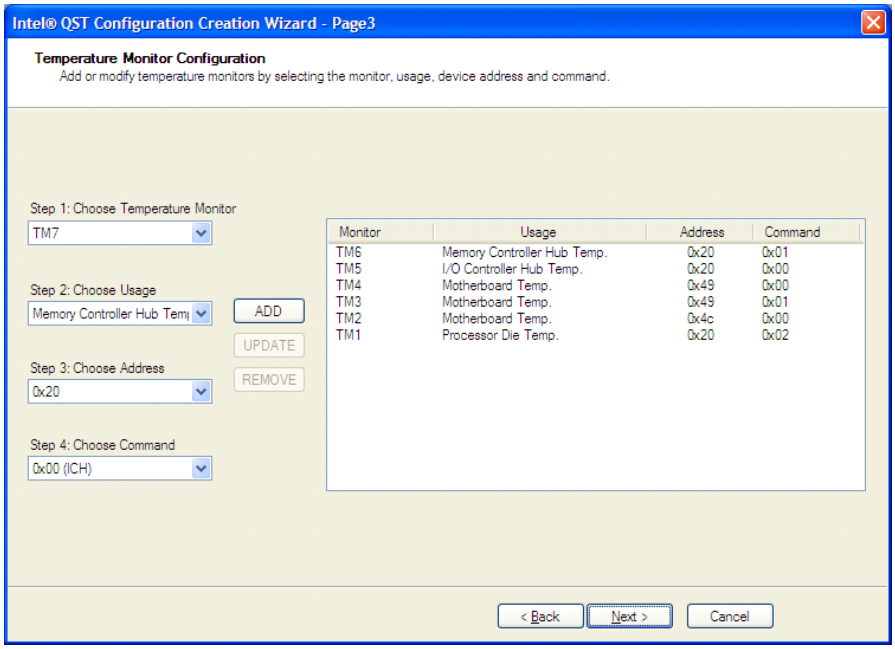
The following figure shows Page 2 of the Wizard. Page 2 prompts the user to add or remove SST devices. If no SST devices are available, checking the check box in the upper left hand corner will disable the page. On configurations where no prior ini file is loaded at the start of the wizard, the user will need to uncheck the check box to enable this page and add devices.



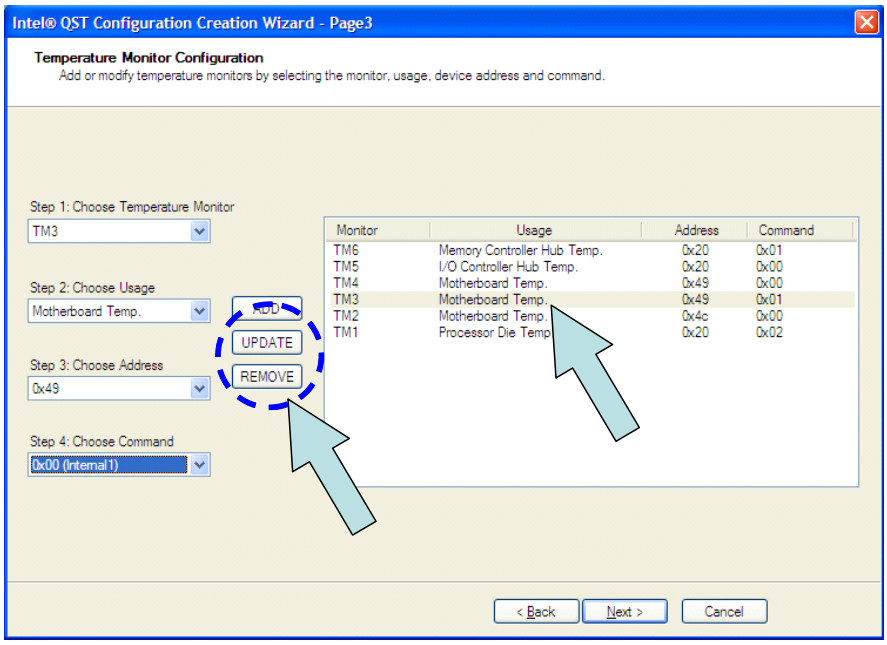
In order to remove a device, the user must first select the device in the list as shown in the following figure to enable the Remove button. Once the device is selected, pressing Remove will delete the device from the list and make the address available in the address window in the lower left hand corner of the page.



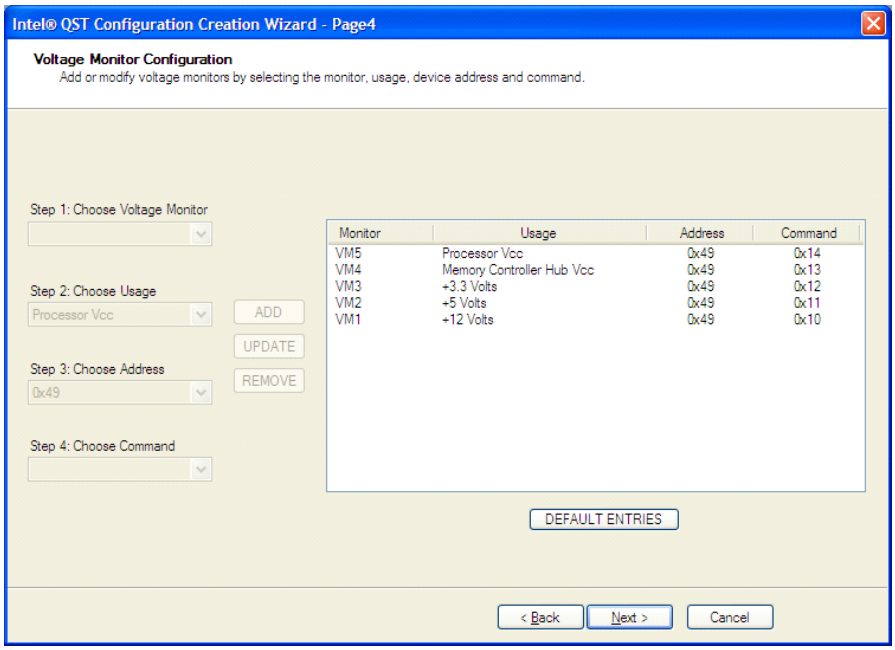
The next figure shows Page 3 of the Wizard. Page 3 configures the Temperature Monitors. Select the options from steps 1 through 4 on the screen and press the Add button to include additional Temperature Monitors. In the event of an error, the Temperature Monitor may be modified by selecting the monitor in the list, making the change, then pressing update.



Note that the Update button will be disabled until a change has been made to the parameters. Likewise the Remove button is also disabled until a monitor is selected from the list.

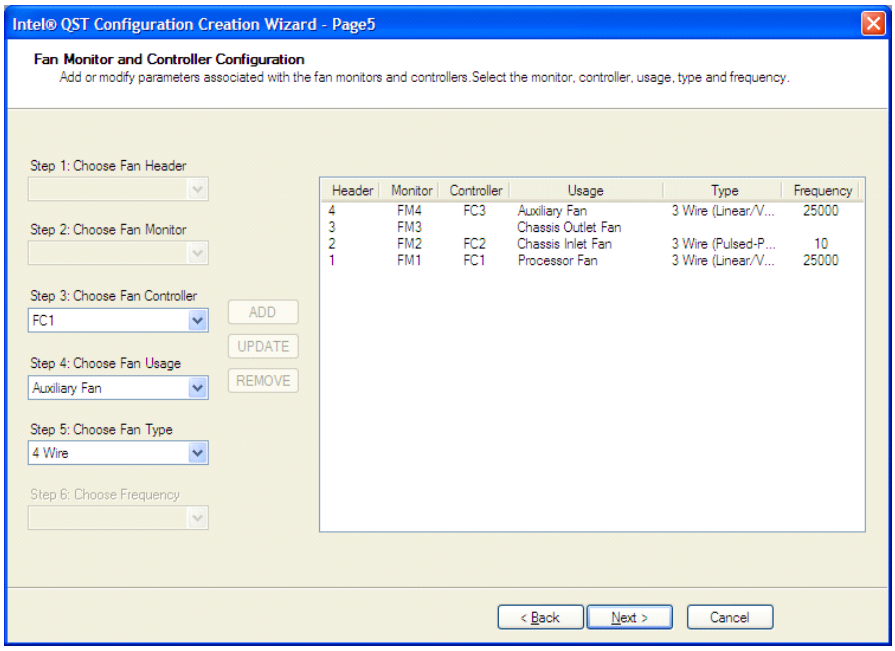


Page 4 of the Wizard is shown in the following figure and configures the voltage monitors. **Note that the Voltage Monitor page will not appear if no SST devices have been added on Page 2.** Adding, modifying, and removing Voltage Monitors are conducted in the same manner as described previously with the Temperature Monitors.

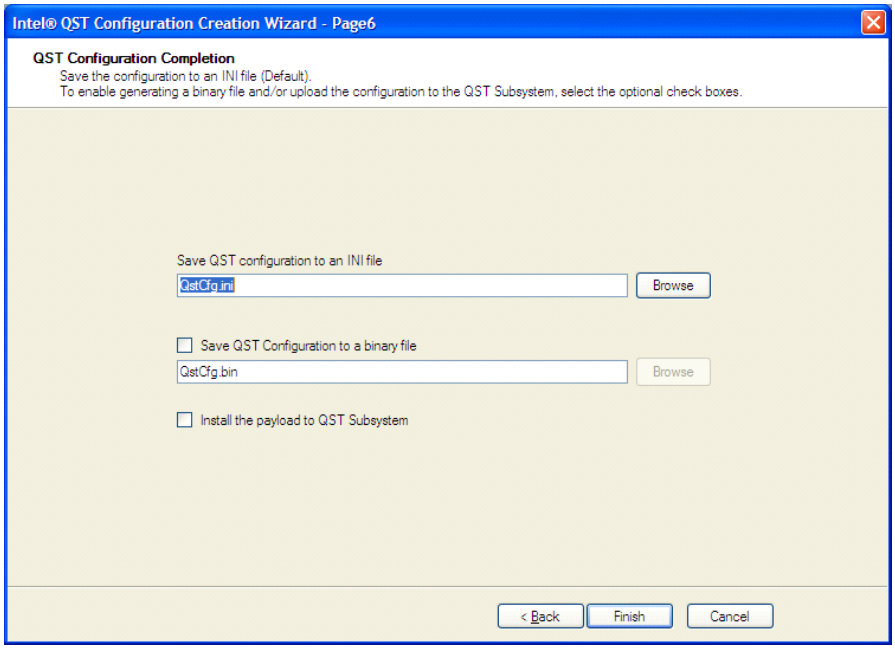


The following figure shows the Fan Monitor and Controller configuration. Note that this page may appear as Page 4 if no SST devices for voltage monitoring have been added. Interfacing with the page is similar to the previous pages. In order to enable

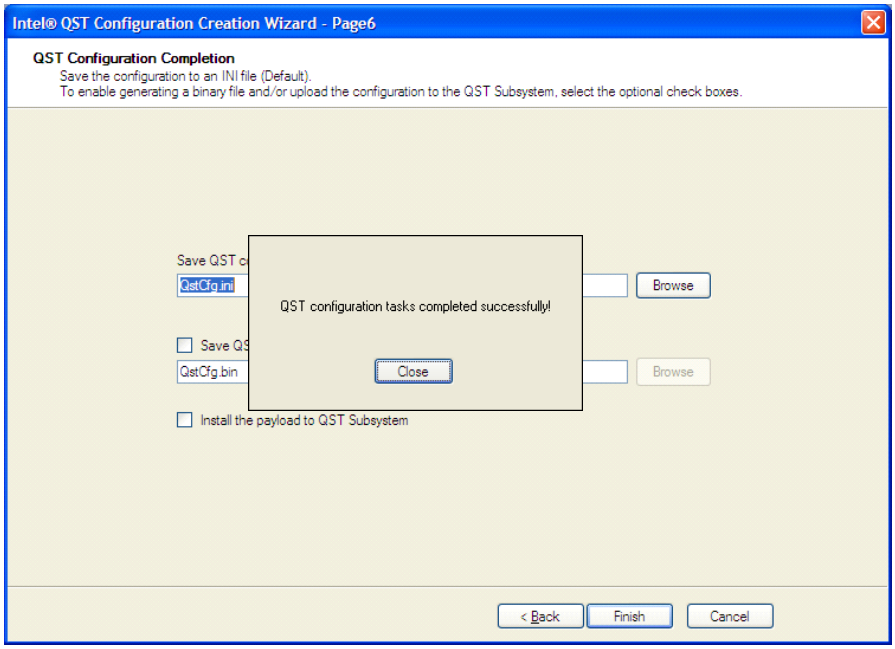
the Update and Remove buttons, one of the monitors must be selected. Also a monitor must be available to configure in order for the Add button to be enabled.



The following figure shows the final screen which may be page 5 or 6 depending upon whether there were voltage monitors for the configuration. The Wizard will automatically create an INI file named QstCfg.ini and save it to the root directory of the Wizard. Select the Browse button to change the destination directory. Two options for the configuration may be selected on this page. A binary image may be generated by selecting the check box. The binary file may be used to build a binary image for flash. If the configuration was conducted on the QST system, selecting the last check box will put the new configuration into the QST subsystem.



Once the destination files are selected, press the Finish button and the Wizard will notify the user of successfully completing the configuration. A dialog window will appear. Clicking Close will close the window and the Configuration Creation Wizard.





3.9 Tuning Wizard

3.9.1 Installation

The Intel® QST Tuning Wizard utilizes the services of a number of other facilities, in order to perform its tasks. As a result of these dependencies, the following installation process is necessary:

1. Install the pertinent version(s) of the Intel® MaxPower Program.
2. Install the Microsoft* .NET Framework.
3. Install the Intel® QST Tuning Wizard.

The Wizard requires Microsoft* Windows XP Service Pack 2* or Windows Vista* operating systems.

3.9.1.1 Installing the Intel® MaxPower Program

The Intel® MaxPower Program is used by the Intel® QST Tuning Wizard to place designated power loads on the processor. There are separate versions of this tool for each individual processor design. You should install versions for each and every processor design that you might utilize in your target system. When you do so, it is required that you install them to their default installation folder (for example: C:\Program Files\Intel Corporation\Maximum Power Program for…); failing to do so will result in the Intel® QST Tuning Wizard being unable to locate and utilize their services.

The executables for installing the MaxPower Program are distributed under the code name of the target processor. Versions for Conroe, Conroe-Lite and Kentsfield are included in Intel® QST Manageability Engine Firmware Kits. They are not included in the Intel® QST Tuning Wizard installation package, however.

3.9.1.2 Installing the Microsoft* .NET Framework

The Intel® QST Tuning Wizard requires that v3.0 of the Microsoft* .NET Framework be installed on the target system. The redistributable package for the framework can be downloaded from http://www.microsoft.com/downloads. Alternatively, the framework is an optional install available via Microsoft Update.



3.9.1.3 Installing the Intel® QST Tuning Wizard

Once you have the Microsoft* .NET Framework and appropriate version(s) of the Intel® MaxPower Program installed onto your target system, you can proceed with the installation of the Intel® QST Tuning Wizard.

To install the Intel® QST Tuning Wizard:

1. Initiate QstTuningWizard.msi.

2. Follow the step-by-step installation instructions. It is recommended that you install the Intel® QST Tuning Wizard to the default directory.

3.9.2 Using the Configuration Wizard

3.9.2.1 Overview

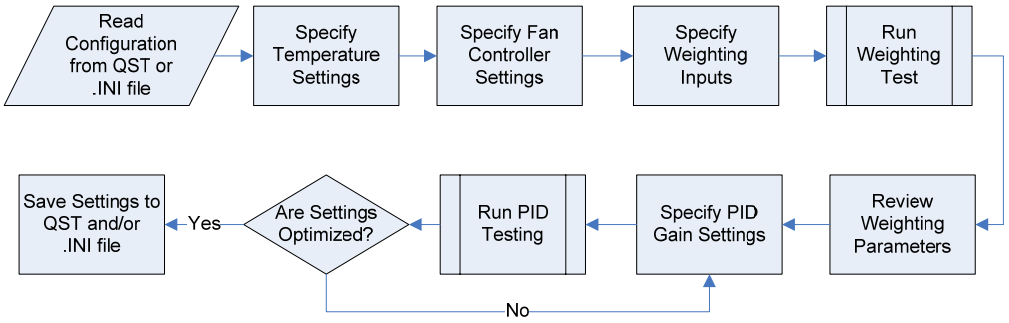
The Intel® QST Tuning Wizard is a simplified user interface that provides the ability to find and set optimal tuning parameters and related configuration parameters. The functionality does not completely replace the other available Intel® QST tools, but provides a tool that will satisfy the System Integrator's basic needs for finding the optimal parameter settings.

The Intel® QST Tuning Wizard is a simplified integration tool to be used by System Integrators to adjust the base configuration for their system designs and automate the tuning of the fan speed control solution. Specifically, it allows the Integrator to set temperature limits, specify fan control settings, specify acoustic information on fans, automatically test and determine temperature-sensor-to-fan-weight parameters, specify and test the PID gain settings for the CPU and deliver the new settings to the Intel® QST subsystem and (optionally) write them to an output (.INI) file.

3.9.2.2 Intel® QST Tuning Wizard Usage Process

The process for using the Wizard and achieving the optimal settings for your system are shown in Figure 1.

Figure 1 - Flowchart of Wizard Process



The Wizard steps through ten screens to complete the tuning process. Each screen and the options available are presented below. For more detailed information, see the Intel® QST Configuration and Tuning Manual.

1. **Welcome:** Start-up screen.

2. **Tool Options:** Used to specify initialization and data-recording parameters for the Wizard.

   - **.INI File:** By default, the active configuration is uploaded from QST and loaded into the current settings fields of the Intel® QST Tuning Wizard. Checking this checkbox allows the Integrator to use the File Open dialog to specify that an alternate configuration should be read from a particular .INI file and used in place of the current Intel® QST configuration.

   - **Log File:** By default, this checkbox is checked and the Intel® QST Tuning Wizard records data to the file "logfile.txt", located in the directory where the Wizard was installed. Clicking the 'Browse' button allows the Integrator to use the File Open dialog to choose the folder and specify the name of the log file that will receive test data (temperatures, duty cycles and fan speeds). Unchecking the checkbox disables the logging of test data.

3. **Temperature Settings:** Used to configure the temperature monitor usage and the associated limit temperature.

   - **Enabled Sensor**: Checkboxes allow the Integrator to disable the use of particular temperature inputs. The Processor, G/MCH and ICH temperatures are of prime importance and cannot be disabled.

     Note: Disabling is a destructive process; if you exit from the Intel® QST Tuning Wizard, the configuration parameters for the temperature input will be lost and you will not be able to re-enable the temperature without directly modifying the .INI file and updating QST. See the Intel® QST Configuration and Tuning Manual for more information.



- **Usage**: Specifies the source of the temperature that is associated with a monitor. For specific devices (Processor, G/MCH and ICH), support for modifying the usage indicator is disabled. Usage for other sensors may be modified as appropriate.

- **Limit Temperature:** Specifies the limit, or target, temperature that Intel® QST should attempt to maintain at the associated temperature sensor.

4. **Fan Controller Settings:** Used to configure the fan controllers that are responsible for the determination of the duty cycle values that are sent to the fans.

   - **Controller:** Indicates the available Fan Controllers, as specified in the current Intel® QST Configuration or the specified .INI file.
   - **Usage:** Specifies the purpose for the fan controllers. The Integrator may change the usage to reflect the most accurate purpose for each fan controller.
   - **Mode:** Specifies the mode of operation for each fan controller. Two modes are supported: **Min** and **Off**.
     - In **Min** Mode, if the necessary duty cycle falls below the specified **Min** duty cycle, the duty cycle is held at this minimum and the fan(s) will remain spinning.
     - In **Off** Mode, if the necessary duty cycle falls below the specified **Min** duty cycle, the duty cycle will be set to 0 (zero) and the fan(s) will be stopped. Once stopped, the fan(s) will remain stopped until the necessary duty cycle exceeds the specified **On** duty cycle.

       Note: **Off** Mode is not supported and is disabled for Fan Controller 1 (typically the processor fan controller).
   - **Min Duty Cycle**: Specifies the minimum duty cycle allowed for the specified fan. The Integrator can change this value if the fans that they are using either do not spin reliably at the current minimum setting or do not meet the idle airflow requirements.
   - **On Duty Cycle**: Specifies the duty cycle above which, if the fans are stopped, they are restarted. This parameter is ignored (and disabled) if operating in **Min** Mode. Specifies the duty cycle at which, if the fans are stopped, they are restarted. The value must be some amount higher than the **Min** Duty Cycle. The difference should be at least 2%, in order to ensure that the fans are not rapidly oscillating between the on and off states.
   - **Max Duty Cycle**: Specifies the maximum duty cycle allowed for the fan controller. This is typically 100%, but may be set to a lower percentage if there is a duty cycle range in which the fan(s) produce significantly higher acoustics without providing any significant improvement in cooling.
   - **Signal Frequency**: Specifies the frequency of the PWM output signal from the fan speed controller. For 3-wire fans, values from 10-94 Hz are typical, though new circuit designs are available that use frequencies in the 22-30 KHz range. For 4-wire fans, 22-30 KHz is typical.

5. **Fan Monitor Settings:** Used to configure the operation of the fan monitors. The fan monitors are responsible for the determination of the health of the underlying fan sensor and the exposure of RPM readings from these sensors.

- **Fan:** Indicates the available fan monitors, as specified in the current Intel® QST Configuration or the specified Intel® QST .INI file.

- **Usage:** Specifies the purpose for the fan that is being monitored. The Integrator may change the usage to reflect the most accurate purpose of the fan monitor.

6. **Weighting Inputs:** Used to provide both required and optional data that is used in the weighting matrix optimization process. Fan acoustic data is included in the weighting matrix calculation in the next step. Intel® QST will de-prioritize louder and less-effective fans in favor of quieter and more-effective fans, respectively, to maintain the minimum system acoustic level.

   - **Ambient Temperature: S**pecifies the ambient temperature of the test room. This is a required parameter.

   - **Acoustic Units:** Specifies the acoustic units of the fan acoustic data, if available. Four choices are possible:

     - **No Acoustic Data:** No data is available to use in this process.
     - **Sound Power:** Acoustic data was measured in BA.
     - **Sound Pressure:** Acoustic data was measured in db.
     - **Relative Loudness:** While acoustic measurement is preferred, subjective evaluation is acceptable. Selecting this option opens the Intel® QST Tuning Wizard fan test window, which allows the Integrator to take the fan controllers to 100% duty cycle in succession, keeping all other fan controllers at their respective minimum duty cycle setting. The Integrator can then subjectively assign the loudest fan controller a 100% relative loudness value and the other fan controller(s) appropriately smaller relative loudness value(s).

   - **Acoustic Inputs:** Specifies the measured acoustic level associated with each fan controller (in units chosen above).

7. **Weighting Calculations:** Used to start the weighting calculation process.

   - **Calculate Weights button:** This button starts the weighting calculation process. Once started, this button can also be used to abort the process (its label will change to depending on the program state). This test finds the optimal power setting, and then sequentially tests the effectiveness of each fan to each temperature sensor.

     Note: This process may take 20-30 minutes to complete, depending on the system configuration, number of fans, etc.

     Note: For best results, the tests should be run with no other programs active. The results depend heavily upon seeing constant power levels, generated specifically by the MaxPower program, and additional CPU load may alter results.

     Note: Some systems will either timeout or be over-temperature even at the lowest MaxPower setting. In the case of a timeout, it is recommended that the Integrator ignore the error and continue the test. If an over-temperature error is returned, the Integrator may choose to continue the test. If they do so, it is recommended that they monitor the system



closely and abort the test if an excessive over-temperature situation occurs. In either case, the Integrator must ensure that each temperature monitor reaches steady-state for optimal weights to be calculated.

8. **Weighting Calculation Results:** Allows user to view and modify the calculated weighting parameters. These parameters are presented in a matrix. The active cells of this matrix can be edited to change these values; however, it is recommended that the calculated values not be modified.

9. **PID Gain Testing:** Used to set, adjust, test and retest the PID gain settings for the processor. The selection of the PID (K_P, K_I, K_D) gain settings is simplified using the Damping and Response parameter table. By analyzing the data created in this step, the gain settings can be adjusted to minimize the transient acoustics on your system. This step is iterative and will likely require the gain testing to be run multiple times.

    - **Gain Presets:** A pre-defined matrix of sets of PID gain settings that will provide good results across a broad range of systems. The default gain setting is (Damping = 1.0, Response = 1.0); this is the recommended settings for starting the test the first time. The corresponding P, I and D gain settings are displayed to the right of the slider bars below.

    - **Maximum Temperature Overshoot:** The maximum difference between the average steady-state condition and the peak across the response time domain. This parameter is displayed for the most recent gain test.

    - **Maximum PWM Overshoot:** The maximum difference between the average steady-state condition of the fan and its peak across the response time domain. This parameter is displayed for the most recent gain test.

    - **Test Gains:** Starts the gain testing routine, which takes the processor through three distinct power steps: from idle to high, from high to mid, and finally, from mid to high (Figure 2, below). By analyzing the fan and temperature response at each of these power steps (marked by a gray vertical line on the plot), the gain settings can be adjusted to minimize the acoustics (see Section 3.3 for guidance on adjusting the gain settings).

    - **Damping:** Damping and Response parameters are a simplified representation of the PID gain settings. In general, increasing the Damping will decrease fan oscillation and decrease temperature overshoot. However, too high of a Damping will cause fan overshoot, causing higher acoustics than are necessary to adequately cool the system.

    - **Response:** Damping and Response parameters are a simplified representation of the PID gain settings. In general, increasing the Response causes the fans to react faster to a temperature change, thereby decreasing temperature overshoot. However, too high of a Response will cause fan overshoot and oscillation, causing higher acoustics than are necessary to adequately cool the system.

    - **Enable Advanced Gain Tuning:** If selected, allows user specify K_P, K_I and K_D independent of the pre-defined ratios. If not selected, K_P, K_I and K_D gain settings are automatically set via the selected radio button in the Gain Presets. Selecting this option gives the user greater flexibility to both have a wider range of gains, as well as independently changing the ratios between

the gains. If settings produce undesirable results, the user can easily select a Damping and Response setting from the pre-defined matrix.

Note: For more information on PID gain setting tuning, see Section 3.3.

10. **Exit Screen:** Allows user to save and implement settings.

    - **Update QST on exit:** Sends the final version of the configuration to QST for storage (overwriting previous settings).
    - **Update .INI on exit:** Writes settings to the .INI file that was specified in the "Tool Options" screen.
    - **Create new .INI on exit:** Writes settings to a new .INI file. The Integrator will specify the filename and directory in a File Open dialogue.

3.9.2.3 PID Gain Setting Tuning

While much of the Wizard is automated, user input is required to find the optimal gain settings. The default gain settings may be adequate for many systems, but fine-tuning the gain settings to your particular system (and especially its processor cooling solution) can provide an additional acoustic benefit. This section describes the process for adjusting the PID gain settings in *Screen 9: PID Gain Testing*.

3.9.2.3.1 Determining the Quality of PID Gain Settings

Certain metrics can be used to measure the quality of your fan speed and temperature response to determine if you have optimized gain settings. Analyzing the fan speed and temperature response to multiple inputs can help you to determine (1) if your current gain settings are appropriate, and (2) how to change the gains to achieve better results. The Intel® QST Tuning Wizard automates both changes in processor power and data collection to enable the Integrator to assess the response quality to changes in power.

3.9.2.3.2 Assessing Response Quality

The first measures of response quality are indicated in the upper right corner of *Screen 9: PID Gain Testing*: the Maximum Temperature Overshoot and the Maximum Fan Overshoot. If either of these values are large (>4 °C or > 5% PWM, respectively) then the user should change the gains and retest. See Table 1 (below) for recommendations on changing the gain settings.

The second measure is analysis of the temperature and fan PWM response data from the PID gain settings test. Figures 3-5 below show the response of a relatively simple one fan, one sensor system while subjected to a series of changes in processor power, shown in Figure 2. The power changes, shown in Figure 2 and moving from left to right, are as follows:

1. Start power at idle setting
2. Increase power to a high power setting
3. Reduce power to an intermediate power setting
4. Increase power to the high power setting again.

This series of power changes will stress the fan speed control subsystem and expose any problems with its dynamic control. In particular, the response at power steps 1-to-2 and 3-to-4 will provide the response data of interest. The results (Figure 3-Figure 5) can be used to illustrate how to measure the quality of fan and temperature response. Table 1 explains the metrics shown in Figures 3-5.

Figure 2 - Power Level Changes for the PID Gain Settings Test

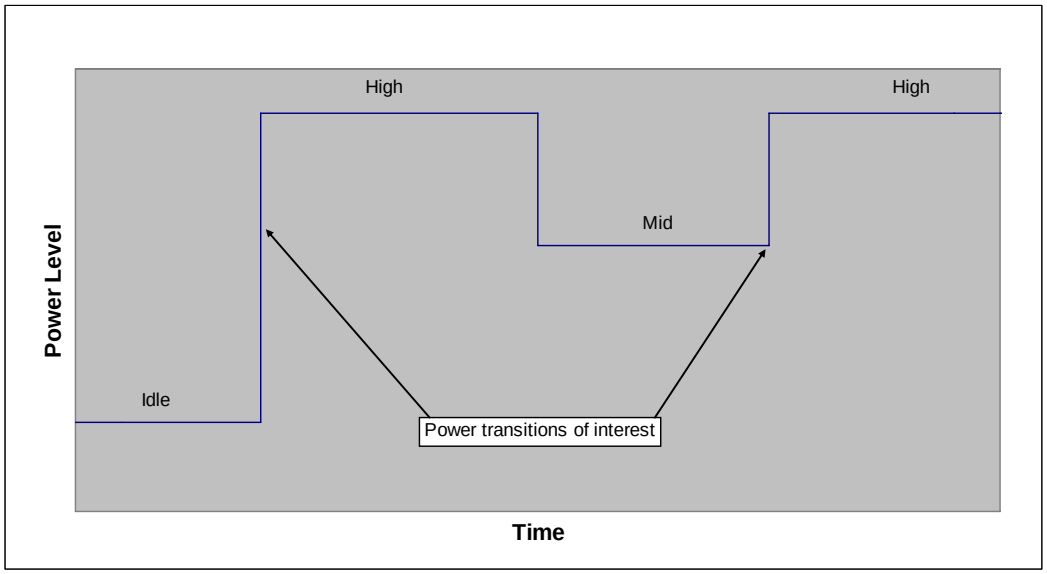


Figure 3, Figure 4 and Figure 5 show response data from a real system with only one fan and the CPU temperature response shown. The indicators of poor PID Gain Settings are highlighted and an improvement path is suggested.

Figure 3 - Fan Overshoot & Temperature Oscillation (Adjustment: Decrease Response)

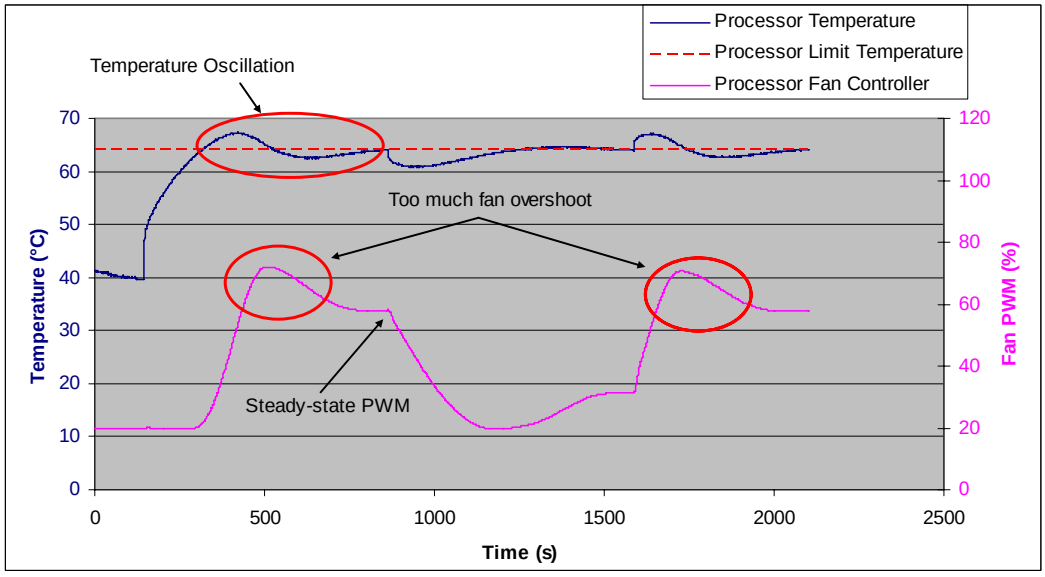


Figure 4 - Temperature Overshoot (Adjustment: Increase Both Damping and Response)

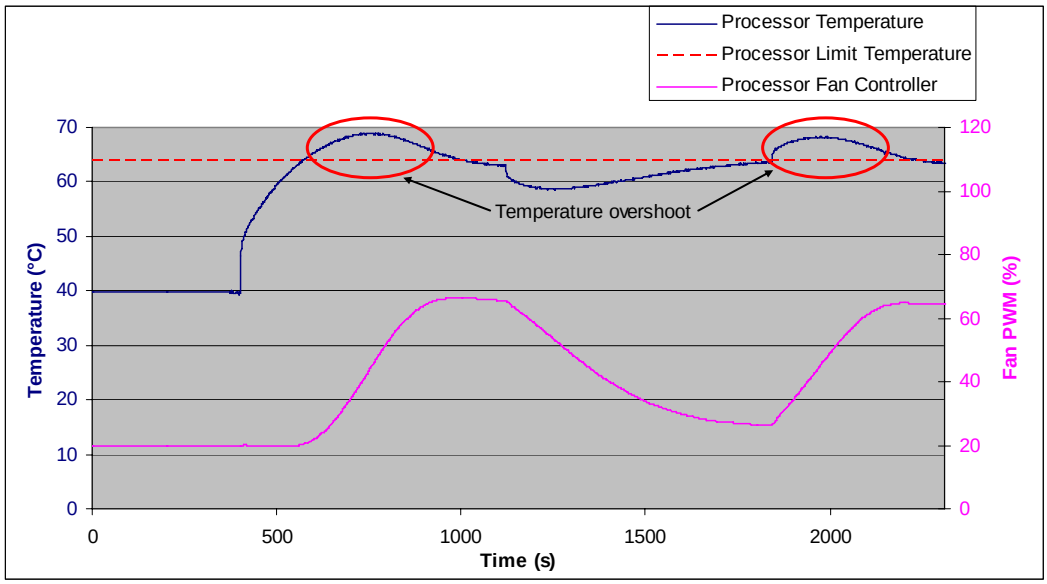


Figure 5 - Fan Spike, Temperature and Fan Overshoot (Adjustment: Decrease Damping)

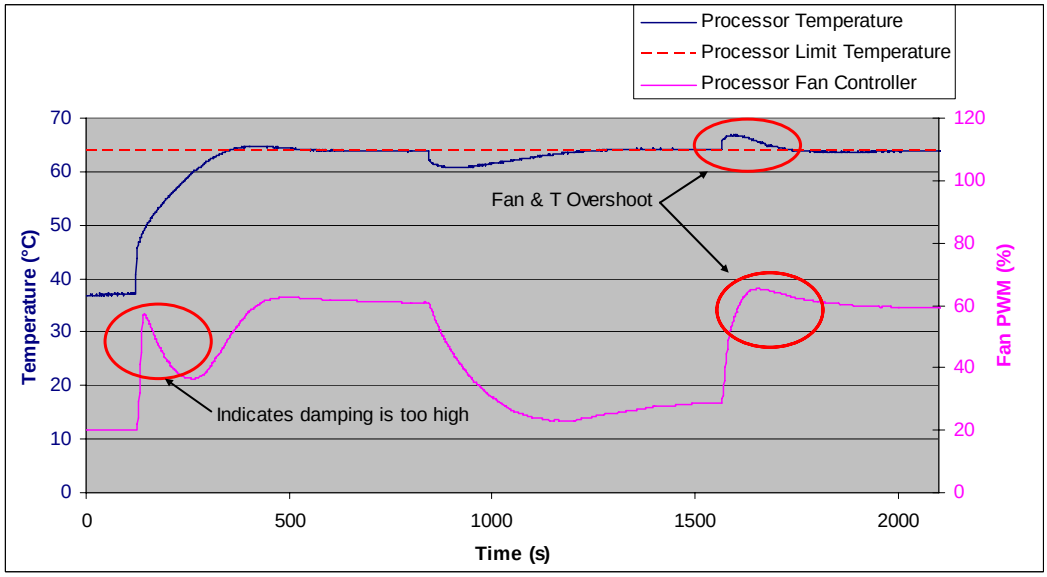




Table 1 - PID Gain Setting Adjustment Guide

Metric	Description	How to Improve Problem
Fan and/or Temperature Oscillation	The fan or temperature response oscillates above and below it's steady-state condition when the power is not changing	Decrease Response and Increase Damping
Fan Overshoot	The maximum delta between the average steady-state condition of the fan and it's peak across the response time domain	Decrease Response and Damping
Temperature Overshoot	The maximum delta between the average steady-state condition of the temperature sensor and it's peak across the response time domain	Increase Response and Damping
Fan and Temperature Overshoot	During certain power changes the fan speed increases above its steady state condition, but at other parts of the test, the temperature increases above its steady state condition	Increase Response and decrease Damping

3.9.2.3.3 Process for Adjusting the PID Gain Settings

Once the response data is analyzed and the problem areas identified, the Damping and Response parameters can be changed to improve performance. The process for tuning the gain settings is as follows:

1. Choose a set of gain values for initial testing from the Gain Preset table. The default settings are a good place to start.

2. Run the gain test on the Processor Gain Tuning screen.

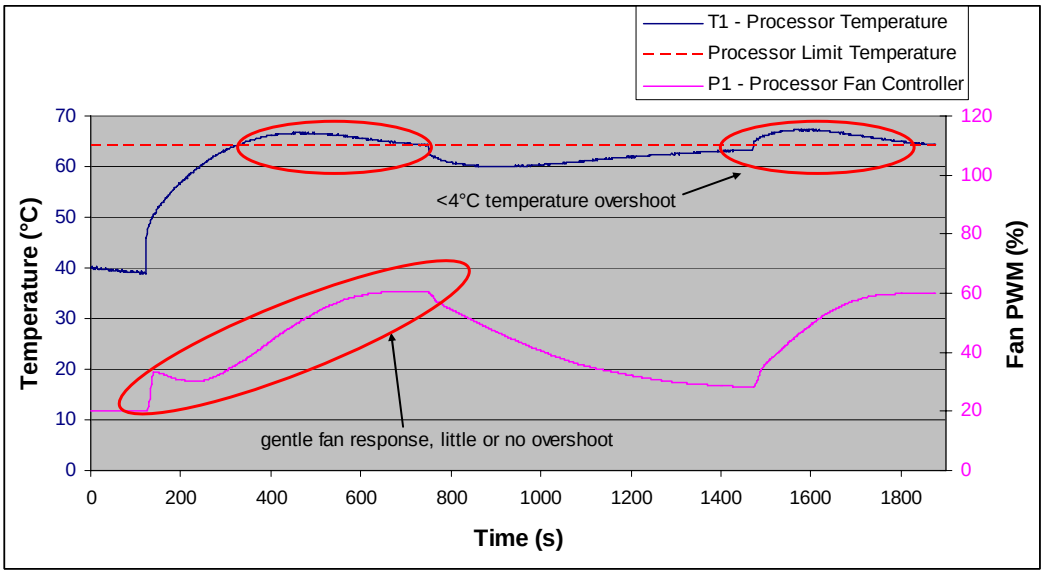
3. Analyze the data and identify the key indicators as described in *Section 3.3.2 Assessing Response Quality.*

4. Use the information presented in Figure 3 through Figure 6 and Table 1 to determine if the PID gain settings are adequate and, if not, how to adjust the Damping and Response to improve the results. For example, if the test was run using the default settings, and the results in Figure 4 were achieved, the temperature overshoot would need to be decreased. Referring to Figure 7, this response indicates moving both the Damping and Response parameters up and to the right is necessary (i.e. away from the 'Too Much Temperature Overshoot' corner). Thus, the button at (Damping = 1.5, Response = 1.5) should be selected. After making any change, the Gain Tests should be run again to verify the new settings.

   Note: This method works for any starting point. Suppose a second test was run and the data indicated that there was too much fan oscillation. For fan oscillation, the plot recommends moving down and right. When the starting point is (1.5, 1.5), the next setting to test would be (2.0, 1.0), and so on. It is also acceptable to move horizontal or vertically on the chart if multiple indicators are found.

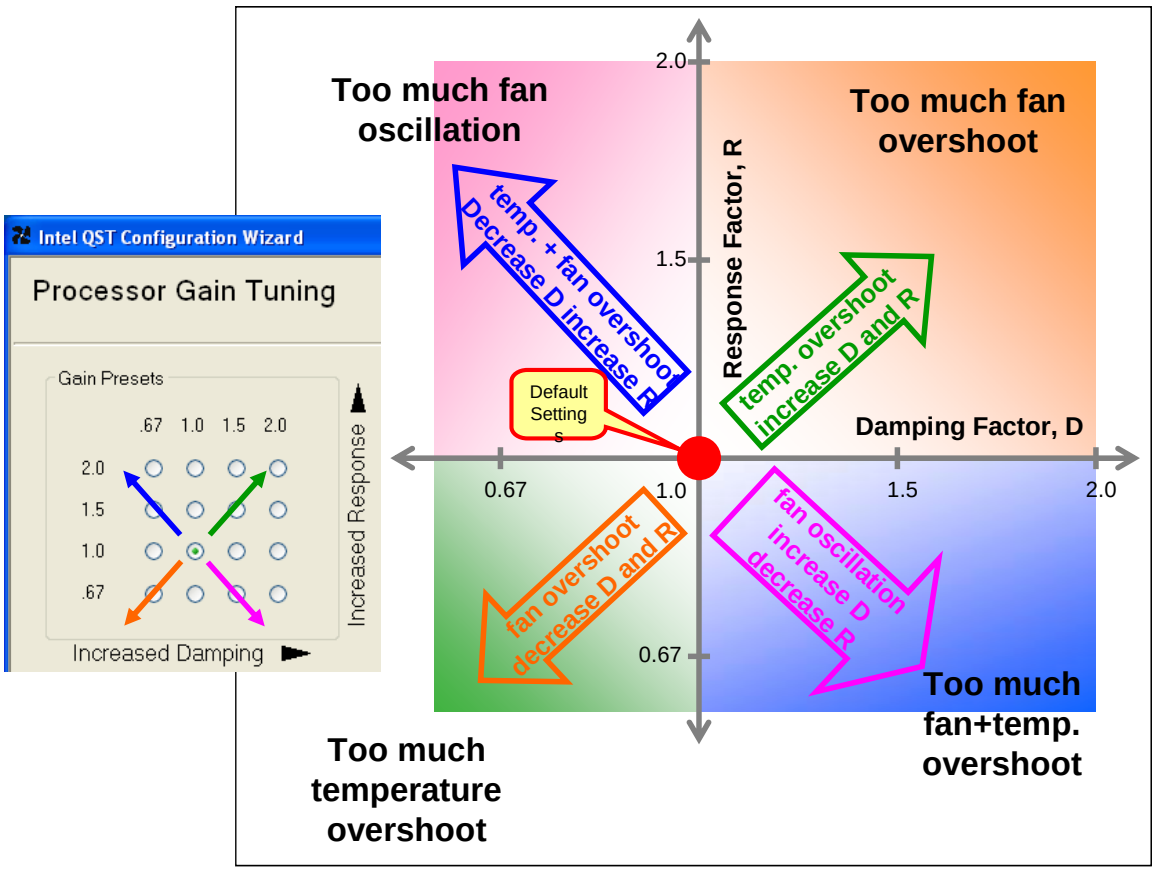
5. Repeat the test, analyze and adjust process (steps 1-4) until good Damping and Response settings are found. This is indicated by none of the symptoms in Table 1 being present in the data, a temperature overshoot of <4°C and a fan overshoot of <5% PWM. Figure 6 shows the response of a well-tuned system.

Figure 6 - Response from Tuned PID Gain Settings



Intel Confidential

Figure 7 - PID Gain Setting Adjustment Guide



3.9.2.4 Using the Plot Window

The QST Plot window offers several functions to help the user see the data as it is collected in real time and in a useful manner.

3.9.2.4.1 Plot Window Contents

The Plot Window shows three separate sets of data vs. time while the testing proceeds: 1) the temperature for each enabled temperature sensor, 2) the fan controller PWM, and 3) the fan monitor RPM. These are displayed top-to-bottom in the window.

3.9.2.4.2 Plot Window Options

The features enabled in the plot window are as follows:

- For both the x- and y-axis:
  - Can be panned by grabbing and dragging.
  - Right click context menu can enable/disable tracking and zoom to fit the data displayed.
- Zoom in on data. To zoom, click the ‘Zoom On’ button in the lower right side of the plot of interest, then select a rectangle around the data of interest. Cursor must start in the upper-right corner of the rectangle. Click the button again (now titled ‘Zoom Off’) to return to the full data set.
- Hide and unhide channels. Click on the channel of interest in the legend and answer ‘Yes’ in the dialogue box to hide a channel. Repeat process to unhide channel.

  Note: Clicking on ‘No’ when hiding or unhiding a channel will open a color selection dialogue to change the color of the channel.
- The window can be resized as desired. Each plot will take 1/3 of the height and the full width of the window.