The Intel® Processor Diagnostic Tool in the Fedora® Live OS - User Guide
LEGAL INFORMATION
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.
UNLESS OTHERWISE AGREED IN WRITING BY INTEL, THE INTEL PRODUCTS ARE NOT DESIGNED NOR INTENDED FOR ANY APPLICATION IN WHICH THE FAILURE OF THE INTEL PRODUCT COULD CREATE A SITUATION WHERE PERSONAL INJURY OR DEATH MAY OCCUR.
Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.
The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request. Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.
This document contains information on products in the design phase of development. This document contains information on products in the design phase of development. Do not finalize a design with this information. Revised information will be published when the product is available. Verify with your local sales office that you have the latest datasheet before finalizing a design.
Code names featured are used internally within Intel to identify products that are in development and not yet publicly announced for release. Customers, licensees and other third parties are not authorized by Intel to use code names in advertising, promotion or marketing of any product or services and any such use of Intel's internal code names is at the sole risk of the user.
Intel and the Intel logo are trademarks of Intel Corporation in the U.S. and other countries. *Other names and brands may be claimed as the property of others. Copyright © 2009, Intel Corporation. All rights reserved.
# TABLE OF CONTENTS

Revision History .................................................................................................................. 4
1 Purpose ............................................................................................................................... 5
2 IPDT Test System Requirements ...................................................................................... 5
3 Acronyms ........................................................................................................................... 6
4 What You Will Need .......................................................................................................... 7
5 Creating a bootable USB flash device ............................................................................. 8
6 Using the LiveUSB .......................................................................................................... 16
7 Installing IPDT ................................................................................................................ 18
8 Running IPDT .................................................................................................................. 20
9 IPDT Configuration .......................................................................................................... 23
10 IPDT Features and Parameters ..................................................................................... 28
## Revision History

<table>
<thead>
<tr>
<th>Revision</th>
<th>Date</th>
<th>Description / Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>2/13/2009</td>
<td>Initial Release</td>
</tr>
<tr>
<td>2.0</td>
<td>10/1/2010</td>
<td>Modified section 5 instructions regarding installing and using liveUSBcreator program. Added Appendix A Table</td>
</tr>
<tr>
<td>3.0</td>
<td>10/11/2010</td>
<td>Added statement in section 5 to specify that the format tool and the livecreator tool must be run in as administrator rights in Windows 7 or Vista. Updated Appendix A</td>
</tr>
<tr>
<td>4.0</td>
<td>11/10/2010</td>
<td>Section 7 updated to include new installation instructions, new Java Graphical User Interface</td>
</tr>
<tr>
<td>5.0</td>
<td>12/3/2010</td>
<td>Updated System Requirements to include 64 Bit Operating Systems</td>
</tr>
<tr>
<td>6.0</td>
<td>1/24/2011</td>
<td>Appendix A to show IMC Speed Test is disabled</td>
</tr>
<tr>
<td>7.0</td>
<td>6/3/2011</td>
<td>Updated screen captures in section 7 from XP to Windows 7 Updated section 9 to include the new IPDT Configure program Changed Appendix A to section 10</td>
</tr>
<tr>
<td>8.0</td>
<td>8/5/2011</td>
<td>Updated sections 6 and 7 to support New Gnome 3 Desktop Environment. Also updated sections 8, 9, &amp; 10 to reflect the new test libraries</td>
</tr>
<tr>
<td>9.0</td>
<td>11/3/2011</td>
<td>Updated sections 6, 7, and 8 removing references to Gnome 3 desktop and adding XFCE desktop. Also updated section 10 to include updates to advanced graphics</td>
</tr>
<tr>
<td>10.0</td>
<td>9/12/2014</td>
<td>Updated supported OS as release 20 and MP support for 64 bit. Updated section 5 requirement of 4GB USB Flash Drive and added alternate diskpart format process. Updated section 6 with Hard Drive Installation process. Updated section 10 to reflect new test libraries.</td>
</tr>
</tbody>
</table>
1 Purpose

The purpose of this document is to provide instructions on how to install and configure a Fedora® Linux Foundation LiveOS that includes the Intel® Processor Diagnostic Tool.

The purpose of the Intel® Processor Diagnostic Tool is to verify the functionality of an Intel® microprocessor. The diagnostic checks for brand identification, verifies the processor operating frequency, tests specific processor features and performs a stress test on the processor.

The diagnostic can be configured to execute in one of two modes, presence test mode or verification mode. Additionally, it can be configured to enable (run) or disable (skip) individual default configurations are used for this mode of operation. For more details see section 10: Features and Parameters.

2 IPDT Test System Requirements

Multiprocessor Systems
The Intel® Processor Diagnostic Tool is compatible with multiprocessor systems. It is essential that only one Intel® processor is installed and tested at a time in a system configuration using the 32 Bit Version of Intel® Processor Diagnostic Tool. Intel® Processor Diagnostic Tool 64 Bit Version (MP) does support multiple processors inserted into a multiprocessor system configuration.

Motherboard & Processor
It is essential that the motherboard you use to test your processor is fully compatible with your Intel® processor. Consult your motherboard manufacturer’s support to ensure the motherboard supports your processor. If you are using an Intel® Motherboard please use this utility Intel® Processors and Boards Compatibility Tool.

Motherboard BIOS
It is essential that the motherboards BIOS is at the minimum BIOS revision specified to support your Intel® processor. Consult your motherboard manufacturer’s support to ensure the BIOS revision is at the correct revision.

Motherboard Architecture
IPDT is only compatible with motherboards built using Intel® Architecture.
Over-Clocking
Over-Clocking should be disabled while running Intel® Processor Diagnostic Tool.

Power Management
Some power management features (e.g. Intel SpeedStep® technology) throttle or reduce the operating frequency of components within the system. These types of power management features may result in very low tested frequency results. This does not mean that the processor is operating at degraded performance levels. It means that the enabled power management feature is optimizing the efficiency of the processor, either to save power or reduce heat within the system.

We recommend you disable any power management features such as Intel SpeedStep® technology and configure your system to its optimal power management settings, when running Intel® Processor Diagnostic Tool. For instructions on how to disable these power management features, please contact your system manufacturer.

Operating Systems
The Fedora® version of the Intel® Processor Diagnostic Tool is compatible with the following operating systems:

- Fedora® release 20 32 Bit (Custom Fedora LiveUSB provided by Intel)
- Fedora® release 20 64 Bit (Custom Fedora LiveUSB provided by Intel)

3 Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>Central Processing Unit</td>
</tr>
<tr>
<td>FSB</td>
<td>Front Side Bus</td>
</tr>
<tr>
<td>IMC</td>
<td>Integrated Memory Controller</td>
</tr>
<tr>
<td>IPDT</td>
<td>Intel® Processor Diagnostic Tool</td>
</tr>
<tr>
<td>LF</td>
<td>Fedora® Linux</td>
</tr>
<tr>
<td>MP</td>
<td>Multiprocessor</td>
</tr>
<tr>
<td>MSR</td>
<td>Model Specific Register</td>
</tr>
<tr>
<td>OS</td>
<td>Operating System</td>
</tr>
<tr>
<td>QPI</td>
<td>Quick Path Interconnect</td>
</tr>
<tr>
<td>USB</td>
<td>Universal Serial Bus</td>
</tr>
</tbody>
</table>
4 What You Will Need

A Windows® based computer which will be used to create the LiveUSB. A 4GB USB flash device is used for the bootable USB flash device.

Warning any data on the USB flash device could be lost! To preserve this data, copy it to another device or computer prior to starting this process.

The IPDT distribution zip file includes the following software:
- Latest Fedora® Live ISO (image) file
- Latest Fedora® LiveUSB creator program – Windows® Version
- USB Formatter Utility
5  Creating a bootable USB flash device

Obtain the distribution zip file for latest version of the IPDT for Fedora® and extract the contents to the desktop for now. You can remove these when you are finished.

Format USB Flash device

Obtain a 4GB or larger USB flash device. This device will need to be formatted in order to use it.

Warning any data on the USB flash device will be lost! Copy any data you want to keep to another device or computer to preserve it.

Insert the USB flash device into the system. Make sure it is the only USB device connected. Locate the USB Formatter Utility as shown below.

For Windows Vista® and Windows 7® the USB Formatter tool must be run as administrator. This can be done by doing a right-click on the icon and selecting “Run as administrator”.

Click “Yes” for the user account control dialog box appears. The window below will be displayed
1. Select the USB device in the **Device** drop down menu using the drop down arrow on the right.

2. Select Fat32 format in the **File System** drop down menu to format the USB with a Fat32 partition.

3. Type in a name of your choice in the **Volume Label** space. You can choose any name you want. For the purposes of the document the “TESTUSB” was entered as the name.

4. Select the Quick Format as the format option.

5. Click Start to format the device.

The dialog box shown below will be displayed.
Click “Yes” to proceed. The dialog box below will be displayed when the utility has finished.

![HP USB Disk Storage Format Tool, V2.1.8]

SanDisk Cruzer 8.02 (1908 MB) (E:\)

The type of the new file system is FAT32.

Volume label is TESTUSB.
Volume Serial Number is CEEE-659B.

1,996,455,936 bytes total disk space.
1,996,451,840 bytes available on disk.

4,096 bytes in each allocation unit.
487,415 total allocation units on disk.
487,415 available allocation units on disk.

Click OK, and then click Close button. Now the USB flash device is ready to be configured as a bootable Fedora® USB flash device. Do not disconnect this device from the system yet.
The Diskpart utility provides an alternate method to prepare a USB flash device on systems with Windows 7® or later release.

Open Command Prompt as administrator and type the following commands. **It is important that the correct disk assigned as the USB flash device is selected in Step 3 below as this operation will format the device and all data on the disk will be lost.**

1. diskpart
2. list disk
3. select disk 1
4. clean
5. create partition primary
6. format fs=fat32 quick
7. active
8. assign
9. exit

```
C:\Windows\system32>diskpart
Microsoft DiskPart version 6.1.7600
Copyright (C) 1999-2008 Microsoft Corporation.
On computer: IPDT-PC

Diskpart> list disk
    Disk  Status   Size     Free
     0       Online  128 GB   67 GB
     1       Online  3819 MB  0 B

Diskpart> select disk 1
Disk 1 is now the selected disk.

Diskpart> clean
Diskpart succeeded in cleaning the disk.

Diskpart> create partition primary
Diskpart succeeded in creating the specified partition.

Diskpart> format fs=fat32 quick
100 percent completed

Diskpart successfully formatted the volume.

Diskpart> active
Diskpart marked the current partition as active.

Diskpart> assign
Diskpart successfully assigned the drive letter or mount point.

Diskpart> exit

C:\Windows\system32>
```
Installing the Fedora® Live OS on USB Flash Device

Find the ISO image file on the desktop. For the purpose these instructions, the image file name will be F14-R01-ipdt1.32.1.0-ics-live. The name of the image you will be using will be different.

Find the liveUSB-creator setup executable and double-click on it to install the liveUSBcreator program.

NOTE: The liveUSB-creator tool must be run as administrator.

When the installation is complete, you will be asked whether to start the program. Click on the “No” button. Go to start the program by clicking on Start → Programs → LiveUSB Creator and right-click to run as administrator.

Click the Yes if the user account control dialog box appears and the window below will be displayed or one similar to it will be displayed.

![LiveUSB Creator](chart.png)

Click the Browse button and the window below will open.
Now click on the Desktop to the left to find image file which was previously copied to the desktop.

Double-click the image file to select it.
Click on the drop down arrow on the Target Device and select the USB flash device that was previously formatted (only one device should be shown).

Left-click and hold on the Persistent Storage bar and slide it to the right until the storage size is approximately 500 Mb. It does not have to be exactly 500Mb.

Click the “Create LiveUSB” button to start the process. Once completed it will show 100% and USB flash device is ready to use for IPDT. See figure below.
You can now close this program. Your USB flash device is ready for use.
6 Using the LiveUSB

Note: The Gnome Desktop environment which was previously used has been replaced with the XFCE Desktop Environment.

XFCE Desktop Environment

When the XFCE desktop launches, the Welcome dialog box shown below may be displayed. Click the “Use default config” button.

![Welcome dialog box]

The Applications Menu and the Log Out button are located at the top of the screen to the left. The main menu panel shown below is located at the bottom of the screen.

![Main menu panel]

IPDT can now be installed on the LiveUSB by proceeding to Section 7 of this document. If it is desired the following steps will install the Fedora® OS to a system Hard Drive.

**Warning any data on the system Hard Drive will be lost! Copy any data you want to keep to another device or computer to preserve it. It is highly recommended that only one Hard Drive be installed on the system for OS installation as the Hard Drive will be partitioned and formatted automatically.**

Locate the “Install to Hard Drive” application on the Desktop as shown below.

![Install to Hard Drive]

Right click on the “Install to Hard Drive” icon to open the selection menu as indicated below, and left click on “Execute” to begin the installation.

The installation will perform all necessary steps automatically with default parameters.

Press the Quit button as indicated below at completion of the installation. The system can then be restarted with the LiveUSB device removed and boot to Hard Drive.
7 Installing IPDT

The next step is to install IPDT and accept the license. IPDT cannot be used until this process is completed.

This process will install IPDT on either the LiveUSB or Hard Drive after installation of the OS.

Click on the “Terminal” Icon shown below in the main menu to launch a terminal window.

To install IPDT in Fedora 32 bit OS type the following command in the terminal window and press the enter key;

install32

Note: use all lowercase letters

To install IPDT in Fedora 64 bit OS type the following command in the terminal window and press the enter key;

install64

Note: use all lowercase letters

The installation process will start and the license will be displayed as shown below.
Press the space bar key to read the entire license or press the letter “q” to quit displaying the license and proceed to the license acceptance.

Press lowercase letter “a” to accept the license and install IPDT. Press lowercase letter “x” to decline the license. If you decline the license, the installation will be terminated.

If you press any other key you will be prompted again.

After the license is accepted, the installation will proceed until completion. When the installation is completed the message below will be displayed;

Installation is Complete!

After the installation is done, the two Icons shown below will appear on the desktop. One icon is for running the diagnostic and the other is for launching the configuration file editor.

IPDT is now ready for use on either the LiveUSB or Hard Drive installation of the OS.
8 Running IPDT

Boot the test platform or test station using the LiveUSB. After the desktop environment is displayed locate the IPDT Icon on the desktop.

Click on the IPDT Icon, A terminal window will open and launch the graphical user interface will open and IPDT will run as shown below.

A progress bar is shown across the top of the control window. Also, there is a terminal window with a scroll bar.

IPDT is executing in the terminal window. At the bottom left of the control window is a list of Features which show the tests are enabled in the default configuration.

At the bottom center is a list of the default parameters used during the test.
The version of IPDT and the test time or duration is listed at the top right of the control window.

When IPDT completes running the progress bar will be converted to display either a PASS or FAIL. The PASS screen is shown below.

![PASS Screen](image)

Now there are two new buttons. The button on the left will close IPDT and the button on the right will turn off the power (shutdown) your computer.
At the top of the Window is a Menu Bar. The File Menu allows you to open the TestResults.txt file using Notepad, and allows you to exit IPDT. See below.

Config will open the configuration menu page for editing. See below.

Help Menu includes About, Contents, and Support information. See below.
9  **IPDT Configuration**

Double-click on the IPDT Configuration Icon and the following will be displayed.

![IPDT Configuration Interface](image-url)
To run IPDT from Preset configurations, select “Preset” menu. You have three preset choices – Quick test, Full Functional Test, Burn-In Test.

Quick Test – All features are disabled. Run just Genuine Intel, Brand String and Frequency Test.

Full Functional Test – All features enabled, run test for 4 minutes with default settings.

Burn-In Test – All features enabled, run the test for 120 minutes. Click “Exit Saving Config” to save your settings and then run IPDT.

To change IPDT Testing Modes, select “Advanced Menu” → Test Mode.

From the Test Mode screen you can change the way IPDT handles a CPU Frequency failure – warn only and continue the test or stop on failure and fail.

You can also Auto detect the processors attributes and use these as expected values or use the IPDTCmdConfig.xml file entries as expected values.

Select “Explain” button for a description of each testing mode.
Note:
Selecting “Use Expected settings in IPDTConfig.xml” may cause IPDT to fail if your processor is not included in the IPDTConfig.xml file. If this happens please select “Add CPU Entry” from the main configuration screens “Advanced” menu.

Add CPU Entry
If you select Testing Mode “Use Expected settings in IPDTConfig.xml” and IPDT does not find your processor (Brand String fails) – then use “Add CPU Entry” to update the IPDTConfig.xml file with your processor information.

When all necessary information is entered, click “Add Entry to IPDTConfig” and the IPDTConfig.xml file will be updated.

Please run IPDT again to test your processor using the expected values in the IPDTConfig.xml file.
The Intel® Processor Diagnostic Tool in Fedora® Live OS User Guide
Revision 10.0
Page 26 of 36
Also, when IPDT is launched it will start testing using the default configuration.

After IPDT finishes testing, the Config Menu will be available from the pass/fail screen. Select Config \(\rightarrow\) Edit config and the following will be displayed

**Configurable Features**
This section lists the individual tests which can be enabled (executed) or disabled (skipped).
The individual tests have a check box which is used to enable or disable the particular test.

**Configurable Parameters**
This section lists the parameters which are configurable for a specific test. When a test has been disabled, the parameter is removed from the list.
### 10 IPDT Features and Parameters

<table>
<thead>
<tr>
<th>Test Library</th>
<th><strong>Genuine Intel Test</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
<td>The purpose of Genuine Intel test library is to validate the CPU being tested is a “GenuineIntel” processor. This is done by reading the contents of the EBX, ECX, and EDX registers, and joining the contents together to construct the Manufacturer name to ensure it equals “GenuineIntel”.</td>
</tr>
<tr>
<td><strong>Options</strong></td>
<td>None</td>
</tr>
<tr>
<td><strong>Parameters</strong></td>
<td>None</td>
</tr>
<tr>
<td><strong>Default Settings</strong></td>
<td>N/A</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Test Library</th>
<th><strong>Temperature Test</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
<td>The purpose of Temperature library is to monitor the temperature of the processor. This is accomplished by reading the DTS value of the processor from the MSR, if supported by the processor. The library determines if the DTS sensor is enabled using CPUID instruction. The value of DTS is an offset value measured in degrees C and is not the actual temperature of the processor. The DTS is the delta between the current temperature and the maximum junction temperature of the die (Tj) which is referenced by the processor as 0. Tj values vary among processors and are not read directly from the processor. If the DTS value is supported and enabled, The DTS value is read and translated as “x degrees C from max temperature,” and reported. The initial execution of temperature library and the following message is after the CPU Manufacturer Test Library has confirmed that a Genuine Intel® processor is being tested. Then it is executed after each test library to monitor the temperature of the processor. The final temperature read it reported at the end of IPDT. Both the initial temperature and the final temperature is saved in TestResults.txt file</td>
</tr>
<tr>
<td><strong>Options</strong></td>
<td>TemperatureTest=&quot;Enable&quot;, temperatureTest=&quot;Disable&quot;</td>
</tr>
<tr>
<td><strong>Parameters</strong></td>
<td>Numerical value in Degrees Celcius</td>
</tr>
<tr>
<td><strong>Default Settings</strong></td>
<td>TemperatureTest=&quot;Enable&quot;; TemperatureWarningTolerance= &quot;0&quot;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Test Library</th>
<th><strong>Brand String Test</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
<td>CPU brand string test library determines which Intel® CPU is being tested using one of two modes, presence test mode or verification mode. The mode being used is determined by the SkipConfig parameter in the IPDTconfig.xml configuration file found in the IPDT folder.</td>
</tr>
<tr>
<td>If the value of SkipConfig parameter is set to “Yes” (i.e. SkipConfig=&quot;Yes&quot;), then the presence mode is being used and the test library will report the CPU Brand String extracted from the CPU registers extracted.</td>
<td></td>
</tr>
<tr>
<td>If the value of SkipConfig parameter is set to “No” then the verification mode is being used and the test library will compare what was extracted from the CPU registers to a known list of brand string names (not, including trademarks, etc)</td>
<td></td>
</tr>
<tr>
<td><strong>Options</strong></td>
<td>SkipConfig=&quot;Yes&quot;, SkipConfig=&quot;No&quot;</td>
</tr>
<tr>
<td><strong>Parameters</strong></td>
<td>None</td>
</tr>
<tr>
<td><strong>Default Settings</strong></td>
<td>SkipConfig=&quot;Yes&quot;</td>
</tr>
</tbody>
</table>
### CPU Frequency Test

**Description**

CPU Frequency test library will validate that CPU operating frequency is within test limits of the expected CPU frequency listed in the CPU Brand String for the CPU unit under test. This is done by extracted Brand String from the CPU registers and parsing the CPU Frequency from the string. If the processor supports base clock and APERF & MPERF MSRs, it will use the registers to calculate the CPU Frequency. Otherwise, it will be using the RDTSC instruction to perform a calculation to determine if calculated CPU Frequency is within limits. The limits are set to +/- 5%.

This test can be impacted when the test platform is not configured to meet the system requirements specified in section 2. To handle this, the Site parameter is used by this test to indicate that the test platform meets all the system requirements. This is achieved when the parameter is Site="Depot". Now, if the detected frequency is outside the limits, the test will stop execution, and display following error message

"CPU Frequency Test Failed!!"

When a test platform is not configured per test system requirements specified in section 2 due to over clocking or power management features, the value must be Site="Other". Now if the detected frequency is outside the limits, the test will continue execution, however the following message will be displayed

"CPU Frequency Test Passed with Warning!!"

**Options**

Site="Depot", Site="Other"

**Parameters**

None

**Default Settings**

Site="Other"

---

### Front Side Bus Test

**Description**

The Front Side Bus (FSB) test library tests the FSB frequency for processor which supports the front side bus. Front Side Bus (FSB) test library executes either in presence test mode or verification mode. The mode being used is determined by the SkipConfig parameter in the IPDTconfig.xml configuration file.

If the value of SkipConfig parameter is set to "Yes" (i.e. SkipConfig="Yes"), then the presence mode is being used and the test library will read CPU Model Specific Register and validate that the FSB is within test limits via lookup tables. The limits are currently +/- 2%.

If the value of SkipConfig parameter is set to "No" then the verification mode is being used and the test library will read CPU Model Specific Register and compare against the BUSSpeed parameter listed in the CPU node list section of the configuration file.

**Options**

SkipConfig="Yes", SkipConfig="No"

**Parameters**

None

**Default Settings**

SkipConfig="Yes"
<table>
<thead>
<tr>
<th>Test Library</th>
<th><strong>BaseClock Test</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
<td>The Base Clock Test verifies the accuracy of the base clock used to calculate the</td>
</tr>
<tr>
<td></td>
<td>Integrated Memory Controller (IMC) link speed and the Quick Path Interconnect</td>
</tr>
<tr>
<td></td>
<td>(QPI) link speed for Intel® Core i7, i5 and i3 Processor, Intel® Xeon Processor</td>
</tr>
<tr>
<td></td>
<td>5500, 5600 &amp; 7500 series and Intel® Processors based on Intel® Microarchitecture</td>
</tr>
<tr>
<td></td>
<td>(Nehalem &amp; Sandy Bridge). The test compares the detected value to the expected</td>
</tr>
<tr>
<td></td>
<td>default base clock of 133MHz or 100MHz(Sandy Bridge). The calculated base clock</td>
</tr>
<tr>
<td></td>
<td>value is calculated by using value of the Maximum Non-Turbo Ratio register from</td>
</tr>
<tr>
<td></td>
<td>Model Specific Register, base operating frequency, current operating frequency from</td>
</tr>
<tr>
<td></td>
<td>CPU Frequency Test and expected base clock.</td>
</tr>
<tr>
<td><strong>Options</strong></td>
<td>BaseClock=&quot;Enable&quot;, BaseClock=&quot;Disable&quot;</td>
</tr>
<tr>
<td><strong>Parameters</strong></td>
<td>Tolerance value from 1% - 100%</td>
</tr>
<tr>
<td><strong>Default Settings</strong></td>
<td>BaseClock=&quot;Enable&quot;; BaseClockTolerance=&quot;5%&quot;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Test Library</th>
<th><strong>Quick Path Interconnect QPI Test</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
<td>The Quick Path Interface Test verifies the QPI rate between processors on a</td>
</tr>
<tr>
<td></td>
<td>multi-processor system. Intel® Quick Path Interface (QPI) test library enumerates</td>
</tr>
<tr>
<td></td>
<td>the buses with QPI group function and retrieves the expected QPI rate. The test</td>
</tr>
<tr>
<td></td>
<td>will use the QPI multiplier to calculate the detected QPI rate and it will either</td>
</tr>
<tr>
<td></td>
<td>displays or compares the expected and detected QPI rate.</td>
</tr>
<tr>
<td><strong>Options</strong></td>
<td>QPITest=&quot;Enable&quot;, QPITest=&quot;Disable&quot; QPITest=&quot;DisplayOnly&quot;</td>
</tr>
<tr>
<td><strong>Parameters</strong></td>
<td>Tolerance value from 1% - 100%</td>
</tr>
<tr>
<td><strong>Default Settings</strong></td>
<td>QPITest=&quot;Enable&quot;; QPITolerance=&quot;5%&quot;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Test Library</th>
<th><strong>Floating Point Test</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
<td>The purpose of this test is to perform addition, subtraction, multiplication and</td>
</tr>
<tr>
<td></td>
<td>division, to test how fast the CPU can perform mathematical floating point numbers</td>
</tr>
<tr>
<td></td>
<td>operations, and do a sum comparison to make sure the floating point is giving the</td>
</tr>
<tr>
<td></td>
<td>correct sum value at the end of each cycle.</td>
</tr>
<tr>
<td></td>
<td>The test program will display Million Floating Point Operations per Second (MFLOPS)</td>
</tr>
<tr>
<td></td>
<td>, time started, cycle completed and quantity of errors detected.</td>
</tr>
<tr>
<td><strong>Options</strong></td>
<td>FloatingPoint=&quot;Enable&quot;, FloatingPoint=&quot;Disable&quot;</td>
</tr>
<tr>
<td><strong>Parameters</strong></td>
<td>Time in seconds</td>
</tr>
<tr>
<td><strong>Default Settings</strong></td>
<td>FloatingPoint=&quot;Enable&quot;; FloatingPointTimer=&quot;2&quot;</td>
</tr>
<tr>
<td>Test Library</td>
<td>Prime Number Generation Test</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td>The purpose of this test library is to test how fast the CPU can search for Prime numbers, reported as operations per second. Prime number generation will be based on the Sieve of Eratosthenes algorithm. The test program will test for unsigned integer type and within the numbers of 0-2³² for 32 bit platform. Random comparison will be done to make sure the prime number generated is a valid prime number.</td>
</tr>
<tr>
<td><strong>Options</strong></td>
<td>PrimeNumber=&quot;Enable&quot;, PrimeNumber=&quot;Disable&quot;</td>
</tr>
<tr>
<td><strong>Parameters</strong></td>
<td>Time in seconds</td>
</tr>
<tr>
<td><strong>Default Settings</strong></td>
<td>PrimeNumber=&quot;Enable&quot;; PrimeNumberTimer=&quot;2&quot;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Test Library</th>
<th>Cache Test</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
<td>The purpose of CPU Cache test library is to verify the size of the L1, L2, and L3 Cache, whichever is present. The CPUID instruction is used to read ECX register. The L1, L2, &amp; L3 Cache size is in 1024-byte units. The test retrieves the L1, L2, and L3 Cache size information and compare with entry in configuration file for CPU unit under test. The cache is tested using one either presence test mode or verification mode. Which mode is used is determined by the SkipConfig parameter in the IPDTconfig.xml configuration file. If the value of SkipConfig parameter is set to “Yes”, then the test library will simply report the cache size. If the value of SkipConfig parameter is set to “No” then the test library will verify the L2 or L3 cache size detected against the expected cache size using the L2Cache or L3Cache parameter found in the CPU node list section of the configuration file.</td>
</tr>
<tr>
<td><strong>Options</strong></td>
<td>CacheTest=&quot;Enable&quot;, CacheTest=&quot;Disable&quot;; SkipConfig=&quot;Yes&quot;, SkipConfig=&quot;No&quot;</td>
</tr>
<tr>
<td><strong>Parameters</strong></td>
<td>N/A</td>
</tr>
<tr>
<td><strong>Default Settings</strong></td>
<td>CacheTest=&quot;Enable&quot;; SkipConfig=&quot;Yes&quot;</td>
</tr>
<tr>
<td><strong>Parameters</strong></td>
<td>N/A</td>
</tr>
<tr>
<td><strong>Default Settings</strong></td>
<td>AVX=&quot;Enable&quot;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Test Library</th>
<th>MMX/SSE Test</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
<td>This test detects which MMX &amp; SSE, SSE2, SSE3, SSSE3, SSE4 instruction sets are supported on the processor being tested. If the instruction set is supported then the test will execute all MMX &amp; SSE instructions for the supported instruction sets. The test will display which features were detected and thus tested.</td>
</tr>
<tr>
<td><strong>Options</strong></td>
<td>MMXSSE=&quot;Enable&quot;, MMXSSE=&quot;Disable&quot;</td>
</tr>
<tr>
<td><strong>Parameters</strong></td>
<td>N/A</td>
</tr>
<tr>
<td><strong>Default Settings</strong></td>
<td>MMXSSE=&quot;Enable&quot;</td>
</tr>
<tr>
<td>Test Library</td>
<td><strong>AVX Test</strong></td>
</tr>
<tr>
<td>--------------</td>
<td>--------------</td>
</tr>
<tr>
<td>Description</td>
<td>Advanced Vector Extensions (AVX) is a 256 bit instruction set extension to SSE and is designed for applications that are Floating Point intensive. The purpose of the AVX test is to detect the presence of the following AVX features on your Intel processor – AVX (Advanced Vector Extensions), AES (Advanced Encryption Standard) &amp; PCLMULQDQ (Carry-Less Multiplication of two 64 bit operands). Note: AVX capability is first detected on your Intel Processor and then it is detected if your operating system supports AVX. Operating Systems that support AVX: Windows 7 SP1, Windows Server 2008 R2 SP1, Linux since kernel version 2.6.30.</td>
</tr>
<tr>
<td>Options</td>
<td>AVX=&quot;Enable&quot; AVX=&quot;Disable&quot;</td>
</tr>
<tr>
<td>Parameters</td>
<td>N/A</td>
</tr>
<tr>
<td>Default Settings</td>
<td>AVX=&quot;Enable&quot;</td>
</tr>
</tbody>
</table>
**Test Library** | **Integrated Memory Controller Test**
---|---
**Description** | The purpose of Integrated Memory Controller (IMC) Test Library is to test Integrated Memory Controller functionality which is included in Intel® Core i7 and i5 Processor, Intel® Xeon Processor 5500 series and Intel® Xeon Processors based on Intel® Micro architecture (Nehalem). This library targets the memory controller speed, memory size, and it performs memory stress test.

**SubTest** | Integrated Memory Controller Test - Memory Size Subtest
---|---
**Description** | The memory size subtest uses the MemorySize parameter located in the global parameter section in the IPDTconfig.xml configuration file. This parameter can be used to enable (execute) or disable (skip) the memory size subtest. If the MemorySize = Enable or "" (no value), then the memory is in the first bank only and report it. This is the default value.

**Options** | MemorySize="3GB", MemorySize="512MB", MemorySize="256KB", MemorySize="Enable", MemorySize="", MemorySize="Disable" (values shown as example only)

**Parameters** | Size GB, MB, KB,

**Default Settings** | MemorySize="Enable"; MemorySize=""

**SubTest** | Integrated Memory Controller Test - Memory Stress Subtest
---|---
**Description** | The memory stress subtest reads and writes using sliding zeros, sliding ones moving inversion algorithms. This subtest uses the MemoryStressTestSize parameter located in the global parameter section in the IPDTconfig.xml configuration file. Setting MemoryStressTestSize as a percentage of available free memory will determine how much available free memory is tested during the Memory Stress subtest.

**Options** | MemoryStressTestSize="1%", MemoryStressTestSize="Disable"; MemoryStressTestTime="10", MemoryStressTestTime="300"

**Parameters** | % of total available free memory Size for MemoryStressTestSize;
Time in seconds for MemoryStressTestTime;

**Default Settings** | MemoryStressTestSize="1%"; MemoryStressTestTime="10"
### Intel 6 Series Chipset and Intel C200 Series Chipset Module

#### Description
The purpose of Intel 6 Series Chipset and Intel C200 Series Chipset Module is to display the chipset information such as chipset family, chipset stepping and external devices that are connected to the chipset. External Device Enumeration sub module currently only supports PCI port, PCIe ports, SATA controllers, USB controller, Intel High Definition Audio Controller and Gigabit LAN.

#### Options
- PCHTest="Enable" PCHTest="Disable"

#### Parameters
- N/A

#### Default Settings
- PCHTest="Enable"

#### SubTest
**Intel 6 Series Chipset and Intel C200 Series Chipset Module – Chipset Identification**

#### Description
The chipset identification sub module displays chipset information including product family and chipset stepping.

### Intel 6 Series Chipset and Intel C200 Series Chipset Module – External Device Enumeration

#### Description
The External Device Enumeration sub module display PCI, PCIe, SATA, USB devices, audio codecs and LAN device which connected to the chipset.

#### Options
- PCHEnum="Enable" PCHEnum="Disable"

#### Parameters
- N/A

#### Default Settings
- PCHEnum="Enable"

### Integrated Graphics Device - IGD

#### Description
The purpose of Integrated Graphics Device is to validate that there is a functioning Intel(R) Integrated Graphics Device on the CPU unit under test. Verifying the presence of the Intel(R) IGD and will list information obtained from reading specific registers from the IGD. Information returned will contain the following: VID2, DID2, RID2, GTTMMADR, GMADR, SVID2, SID2, ROMADR, GMS, IVD, and Frame Buffer Size determined from GMS register value.

If IGDTest parameter is set to “Disable” then IGD will not execute and subset tests “GFXTestExe” or “GFX2DTestExe” will not execute regardless of the GFXTest setting.

#### Options
- IGDTest="Enable" or IGDTest="Disable"

#### Parameters
- N/A

#### Default Settings
- IGDTest="Enable"
### Advanced Graphics Testing – GFX Library

**Description**

The purpose of Advanced Graphics Testing is to exercise the GPU by displaying graphics. The graphics library calls a separately developed OpenGL graphics binary that is displayed during test.

- If IGDTest parameter is set to “Enable” and IGD detection is true and GFXTest is set to “Enable” and then the 3D-GFX test will execute.
- If IGDTest parameter is set to “Enable” and IGD detection is true and GFXTest is set to “Disable” then the 3D-GFX and 2D tests will not execute.
- If IGDTest parameter is set to “Disable” then IGD will not execute and the subset test “GFXTest”, “2DTest”, and “Rotation” will not execute regardless of the GFXTest setting.
- If the GFX window is inadvertently closed by the user before the allotted time specified by GFXTime then IPDT will generate a fail.
- If Rotation parameter is set to “Enable” then Rotation of display will execute.
- If Rotation parameter is set to “Disable” then Rotation of display will not execute.

**Options**

- GFXTest=”Enable” or GFXTest “Disable”, GFXTestExe=”visGFX1.exe” for Windows or GFXTestExe=”./visGFX” for Linux, GFX2DTestExe=“vis2Dgfx.exe for Windows or GFX2DTestExe=“./vis2Dgfx” for Linux, GFXTime=”20” or GFXTime=””, GFXSpin=”20” or GFXSpin=””, Rotation=”Enable” or Rotation=“Disable”

**Parameters**

- GFXTime parameter takes in a time value in seconds. This informs IPDT how long to display the visual. If GFXTime is set to “” then default time is 20 seconds.
- GFXSpin parameter takes in a spin speed value. The lower the value, the faster the spin of the visual. If GFXSpin is set to “” then spin value is set to 10.

**Default Settings**

- GFXTest=”Enable”, GFXTestExE=“visGFX1.exe”, GFXTime=“”, GFXSpin=“10”, GFX2DTestExe=“vis2Dgfx.exe”, Rotation=“Disable”

### Multi-core/Multi-threaded Test Library

**Description**

The Multi-core, Multi-thread test library will stress the CPU cores for a given period of time using Intel® Threading Building Blocks. This test uses the StressTestTime and StressLoadLevel parameters both located in the global section in the IPDTconfig.xml configuration file.

- The StressLoadLevel parameter is a numerical value which represent the number of tasks required to sufficiently load the cores.
- The StressTestTime parameter is used to set the duration of this test library, thereby, extending the total test time for the diagnostic.

**Options**

- StressTest=“Enable”, StressTest=“Disable”

**Parameters**

- Number of Tasks; Test Time;

**Default Settings**

- StressTest=“Enable”; StressLoadLevel=“8”; StressTestTime=“4”
### Test Library: Test Results File Name

<table>
<thead>
<tr>
<th>Description</th>
<th>The parameter &quot;OutputResultsText&quot; is used to set the name of the output file</th>
</tr>
</thead>
<tbody>
<tr>
<td>Options</td>
<td>N/A</td>
</tr>
<tr>
<td>Parameters</td>
<td>Name of File</td>
</tr>
<tr>
<td>Default Settings</td>
<td>Default file name is TestResults.txt</td>
</tr>
</tbody>
</table>

### Test Library: Test Results File Format

<table>
<thead>
<tr>
<th>Description</th>
<th>The value of parameter &quot;OutFileFormat&quot; is used to store the test results in text file format or XML file format.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Options</td>
<td>Text / XML</td>
</tr>
<tr>
<td>Parameters</td>
<td>N/A</td>
</tr>
<tr>
<td>Default Settings</td>
<td>Test</td>
</tr>
</tbody>
</table>

### Test Library: Output Pass Text File (OutputPassText)

<table>
<thead>
<tr>
<th>Options</th>
<th>Name of File</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
<td>pass.txt</td>
</tr>
<tr>
<td>Default Settings</td>
<td>Used to support internal legacy test processes. Sets filename flag creation when IPDT passes test</td>
</tr>
</tbody>
</table>

### Test Library: OutputFailText

<table>
<thead>
<tr>
<th>Options</th>
<th>Filename</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
<td>fail.txt</td>
</tr>
<tr>
<td>Default Settings</td>
<td>Used to support internal legacy test processes. Sets filename flag creation when IPDT fails test</td>
</tr>
</tbody>
</table>

*Microsoft, Windows and the Windows logo are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries

**Intel, Intel SpeedStep and the Intel logo are a trademark or registered trademark of Intel Corporation or its subsidiaries in the United States and other countries*