

Intel® Integrator Toolkit Usage Overview White Paper (v2.0.0)

Introduction

Objective

This paper provides an overview of the Intel® Integrator Toolkit, outlining its primary usage model and presenting the steps involved in deploying the toolkit into a manufacturing environment. An extensive help system included with the toolkit details many of the specifics involved in deployment of this powerful utility into a production environment. Supplied with this white paper and the toolkit's help system, an OEM has all the information required to incorporate the Intel Integrator Toolkit into a production environment.

Application Overview

Intel Integrator Toolkit is a utility developed by Intel specifically for OEMs. It streamlines the configuration of system BIOS settings and can be used to verify the configuration of an as-built PC system. The most recent version of Intel Integrator Toolkit, version 2.0, includes the following primary features:

- OEM Specified BIOS Settings: Redefine CMOS defaults shipped from the Intel factory. When end users select F9 from the BIOS setup utility to restore defaults, the OEM defaults are recovered.
- OEM Specified SMBIOS Values: Specify Type 1 (manufacturer information) and Type 3 (chassis information) SMBIOS values that persist in the BIOS ROM.
- Lock or Hide Selected BIOS Settings: Lock or Hide individual BIOS setup options to prevent end users from altering BIOS values. Locking a setting grays the value out in the setup utility so that the end user can see the setting's value, but cannot change it. Hiding the value removes the setting entirely from the setup utility so that it cannot be viewed or changed.
- Replication of BIOS Settings: Install specified BIOS defaults and SMBIOS values onto a system during production with a fast and simple flash update operation.
- BOM Extraction: Extract a bill of materials (BOM) from an existing system. The extracted BOM can be used as a report to document as-built contents of a finished system. It can also be used with the BOM Verify feature to compare any system's contents against a standard reference configuration.
- BOM Verification: Compare any system's contents against a BOM that has been extracted from a physical reference system.

Understanding the Basics of Intel Integrator Toolkit

This section presents the integrator toolkit usage model at its highest level, explains its primary programmatic components, and outlines the different types of files involved in its usage. The section following this one will cover the usage model in greater detail by presenting the usage flow and detailing the specific steps involved in putting the toolkit to work.

Primary Usage Model

Figure 1 summarizes the integrator toolkit usage model at its most fundamental level. There are two primary steps involved.

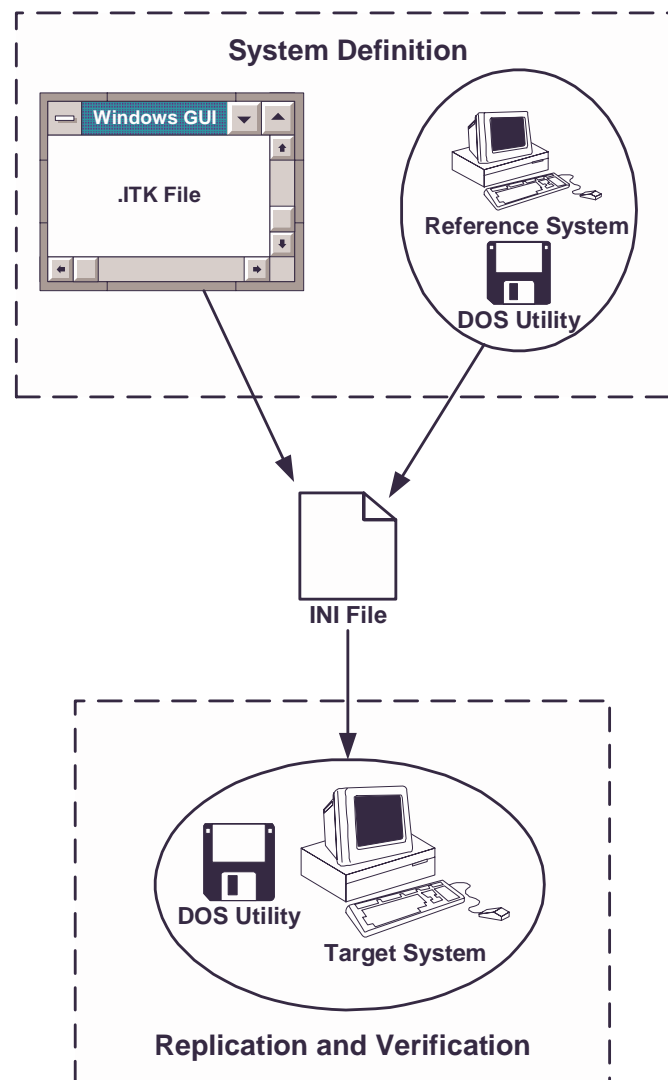
- Step 1: System Definition – Specify BIOS settings for the target system by either:
 - (a) building and configuring a reference system and using the DOS utility to extract systems settings and BOMs, or
 - (b) specifying BIOS settings without a reference system by using the Windows* OS-based

graphical user interface (GUI).

- Step 2: Installation and Verification – Use the DOS utility to install the specified BIOS settings onto a target system and verify the system's as-built BOM against a target BOM extracted from a reference system.

Note: While either the DOS tool or the Windows* OS-based GUI may be used in the configuration of BIOS settings, only the DOS tool can be used to install BIOS settings onto target systems and to conduct BOM operations of any sort.

Figure 1: Primary Integrator Toolkit Usage Model



Program Components

Intel Integrator Toolkit consists of two programmatic components:

- The DOS utility is used to (a) extract custom BIOS settings and the hardware configuration from a physical reference system, (b) install custom BIOS settings onto a target system during manufacturing, and (c) compare a target system's hardware configuration (its BOM) against that of a reference system.
- The Windows OS-based GUI is used to specify custom BIOS settings (without requiring a

physical reference system to be built) and save those settings to an INI file. The INI file is then used by the DOS utility to install the custom BIOS settings onto a target system.

Files

There are four primary file types that the integrator toolkit uses as input and output:

- **BIOS Files** – These are the binary files that compose the BIOS image that is flashed onto a system. They consist of a number of files with extensions of .BIO, .Blx (where x is an integer), and .SIG. The BIOS files, which come packaged as a set specific to a particular platform and BIOS production release, are used by the DOS utility to replicate BIOS settings onto a target system. They are publicly available on the web at:
<http://developer.intel.com/design/motherbd/genbios.htm>.

Note: BIOS files can be downloaded in two formats: as an Intel® iFlash package or as an Intel® Express BIOS Update package. Integrator toolkit uses the iFlash package and *will not* work with the Intel Express BIOS Update package.

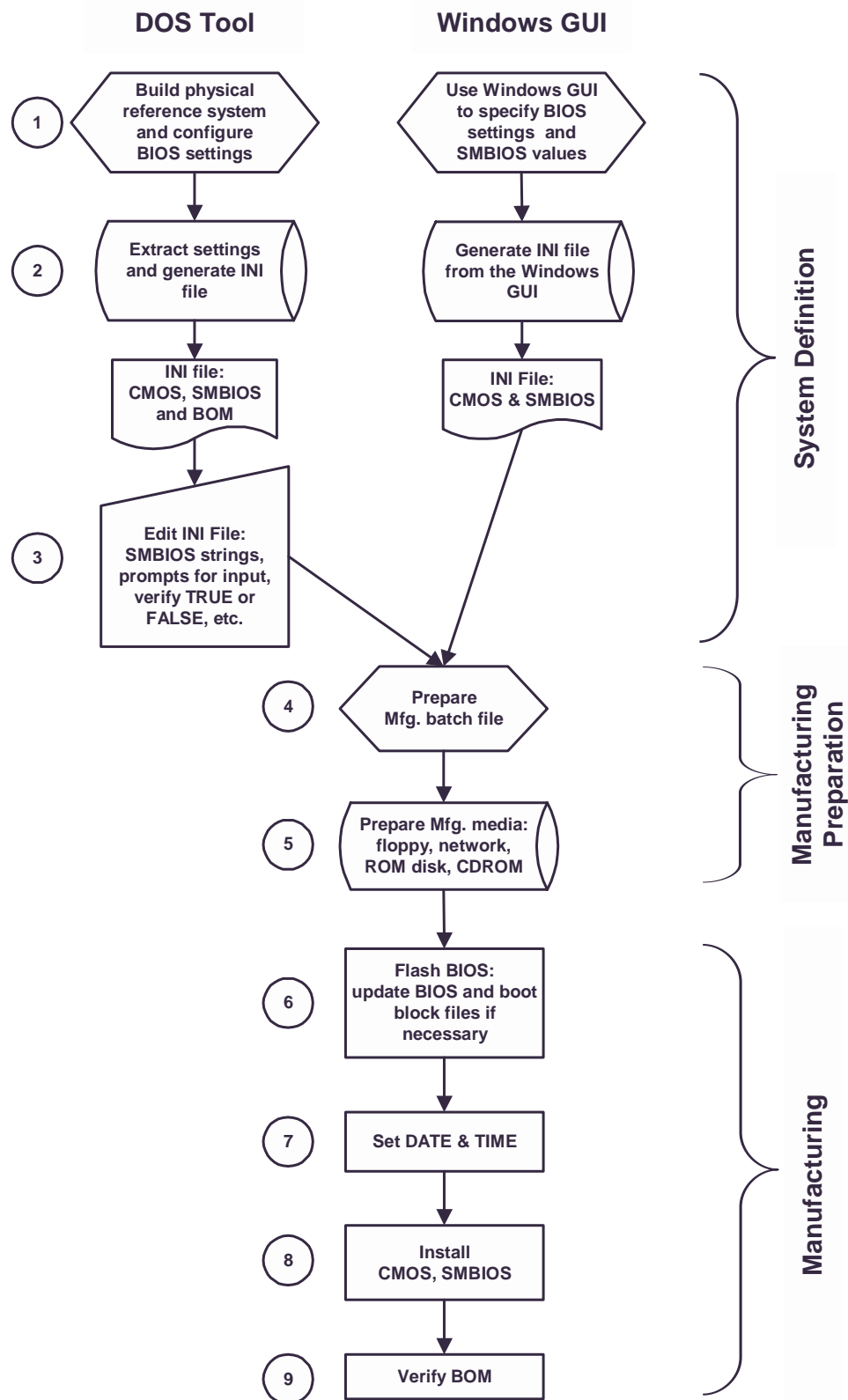
- **BIOS Configuration File** – This file is contained in the iFlash BIOS updated package and is identified by its .ITK extension. The BIOS configuration file is a map of the setup options for a specific BIOS production release (hence, it is sometimes referred to as the “map” file); it allows for the “reference-free” configuration of BIOS settings via the Windows GUI. In order to use integrator toolkit with the Windows OS-based GUI, a BIOS configuration file is required.
- **INI File** – This file, designated by an .INI extension, is the primary output from the system definition process, and the fundamental input to the replication and BOM verification process. The INI file is an editable ASCII file containing a binary block of data that comprises the BIOS CMOS settings.
- **Workspace File** – This file is specific to the Windows OS-based GUI, and is identified by its .IWP extension. It is used to save the entire workspace for a given project so that it can be recalled at a later date to the exact state in which it was saved.

Table 1 outlines the specific functions and file types covered by the DOS utility and the Windows OS-based GUI.

Table 1: Toolkit Files and Functions

	DOS	Windows
Configuration Functions		
Extract CMOS settings from reference system	X	
Specify CMOS settings <i>without</i> a reference system		X
Specify SMBIOS values	X	X
Extract BOM settings from reference system	X	
Lock/Hide BIOS settings		X
Generate INI file	X	X
Installation and Verification Functions		
Install CMOS defaults	X	
Install SMBIOS values	X	
BOM verify	X	
Input and Output File Types		
BIOS Binary Files (*.BIO, *.Bix, *.SIG)	X	
BIOS Configuration File (*.ITK)		X
INI File (*.INI)	X	X
Workspace File (*.IWP)		X

Figure 2: Intel® Integrator Toolkit Usage Model Flow



Putting the Intel Integrator Toolkit To Work

This section outlines the specific implementation steps involved in applying the Intel Integrator Toolkit to a production environment. As outlined in Figure 2 (above), the entire process is divided into nine distinct steps that are grouped into three primary process phases:

- System Definition
- Manufacturing Preparation, and
- Manufacturing.

The following contains an overview of each phase and a discussion of each individual step in the process flow. Where appropriate, references to pertinent sections of the help system are called out with a **Details in Help** notation.

System Definition Phase

The system definition phase entails specifying what the CMOS and SMBIOS settings for the final system will be, and, if using BOM verification, defining the hardware and firmware configuration of the system.

The final output of the system definition phase is an INI file that is used to replicate settings and/or verify the final as-built system configuration. The system definition phase can be accomplished using either the DOS tool or the Windows GUI. The DOS tool method requires that a physical system be built and manually configured, while the Windows GUI method can be accomplished “reference free” without the need for a physical system.

Step 1 – System Configuration (DOS or Windows)

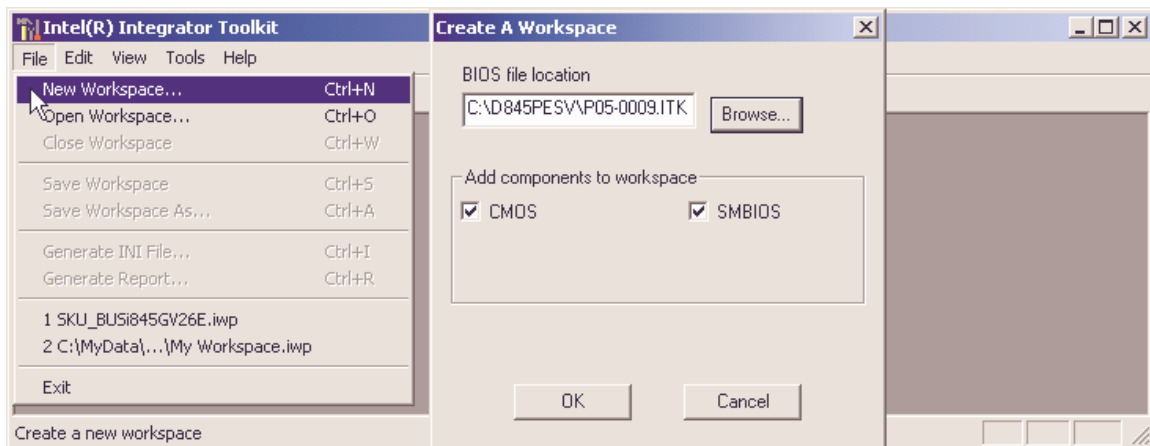
DOS Utility Method

1. Build and configure the physical reference system. This entails adding all necessary hardware devices (PCI, IDE, SCSI, fan, memory, and CPU) to the Intel® desktop board. If the BOM verify feature will be used during the manufacturing process, then the system is likely a prototype that will be cloned on the manufacturing line. In this case, ensure the hardware in the reference system is identical to that desired in the final target system(s).
2. Verify that the desired production BIOS version resides on the system. If necessary, flash the correct version onto the system using either an iFlash bootable floppy, or, if a Windows operating system is installed on the system, Intel Express BIOS Update.
3. Reboot the system, enter the BIOS setup environment by hitting F2 during POST, and customize BIOS settings to the desired configuration. When finished, exit setup and be sure to save changes.

Windows GUI Method

1. Obtain the iFlash BIOS update package for the production BIOS release to be used on finished systems. The latest BIOS updates are publicly available from <http://developer.intel.com/design/motherbd/genbios.htm>. Unpack the contents of the BIOS update package to a location accessible from the system containing the Intel Integrator Toolkit Windows GUI.
2. Create a workspace in the Intel Integrator Toolkit using the .ITK file found in the BIOS update package (see Figure 3 below). Configure BIOS CMOS settings and specify SMBIOS values or SMBIOS input prompts as desired using the Windows GUI. Save the workspace file to a location that can be accessed at a later date, should changes or modifications be necessary.

Details in Help – “Using the Toolkit | Windows-based Interface | Defining My System”

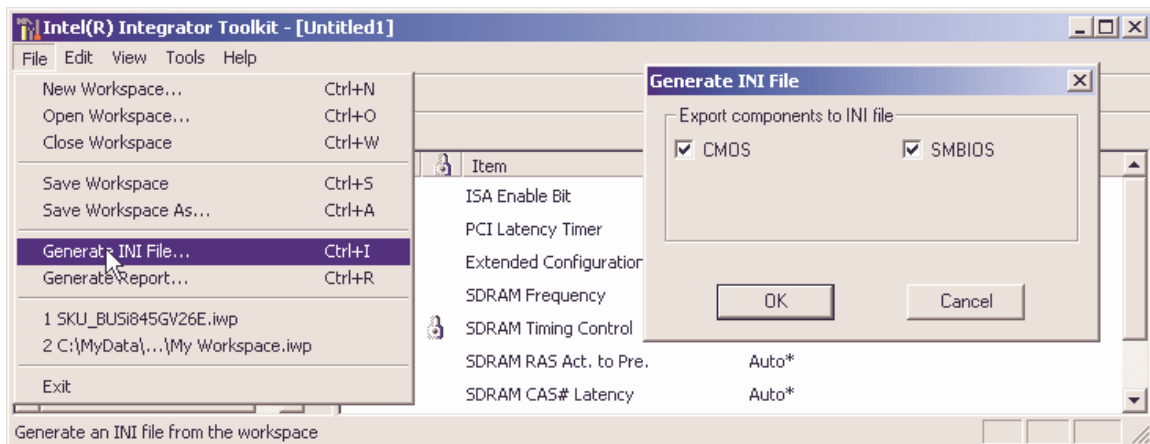
Figure 3 – Creating a workspace and selecting the .ITK file using the Windows GUI**Step 2 – INI File Generation (DOS or Windows)***DOS Utility Method*

Use the iToolkit.exe EXTRACT command to capture all CMOS settings, SMBIOS values and BOM information from the reference system to an INI file.

Details in Help – “Using the Toolkit | DOS-based Interface | Integrator Toolkit Components | Integrator Toolkit Components Overview”

Windows GUI Method

Use the Generate INI function in the Windows GUI to create an INI file with the specified CMOS settings and/or SMBIOS values.

Figure 4 – Generating an .INI file using the Windows GUI

Details in Help – “Using the Toolkit | Windows-based Interface | Defining My System | INI Files | Generate an INI File”

Step 3 – Edit the INI File (DOS or Windows)

The two most common reasons to edit the INI file are:

1. to specify SMBIOS values or prompts, and
2. to apply special conditions when using the BOM verify function.

Since there is no way to define SMBIOS values from a physical reference system, and since BOM extraction can only be accomplished when using the DOS tool with a physical reference system, modification of the INI file is necessary only when using the DOS tool in the system definition phase. While an INI generated from the Windows GUI can be manually edited,

there is rarely reason to do so, since all possible changes can be accomplished through the Windows GUI.

Note: Any modifications to the INI file should be done with care, and only in accordance with the WARNING message that appears at the top of each INI file.

SMBIOS values help identify the system by persisting unique information on the BIOS ROM. When using the DOS tool, Type 1 (manufacturer info) and Type 3 (chassis info) SMBIOS values are manually entered in the INI file as ASCII text strings after the extract operation is conducted. If values are to be entered on the manufacturing line (such as with a keyboard, bar code scanner, or other input device), specify the SMBIOS value \$PROMPT\$ string so that the manufacturing technician will be prompted with a descriptive statement.

Details in Help – “Using the Toolkit | DOS-based Interface | Toolkit Reference and Formats | INI Configuration File Format”

If using the INI file to conduct BOM verification at the end of the manufacturing process, modifications to the BOM portion of the INI file may be desired. Such modifications involve manually setting VerifyExact and Verify values to TRUE or FALSE.

Details in Help – “Using the Toolkit | DOS-based Interface | Integrator Toolkit Components | BOM Verify Component”

Manufacturing Preparation Phase

Manufacturing preparation consists of two steps involving the set up of files and media necessary for system production. Neither of these steps requires direct use of the Intel Integrator Toolkit (except for any possible testing), but are necessary to deploy the toolkit in the manufacturing environment.

Step 4 – Prepare the Manufacturing Batch File

Preparing the manufacturing batch file requires a working knowledge of DOS batch file scripting and an understanding of the features and functions of the Integrator Toolkit. Integrator Toolkit commands can be placed directly in the autoexec.bat file, or in another batch file that is called from the autoexec.bat file. Whatever implementation methodology is used, the full paths and filenames of the INI and BIOS files, as well as the location of Integrator Toolkit executable, must be known.

Step 5 – Create the Manufacturing Media

The bootable media required for manufacturing will vary depending upon the manufacturing environment. In a non-automated, non-networked environment, a floppy diskette may be the most appropriate media, while in a more automated environment that is networked, a bootable floppy or CD-ROM might be used to establish network connections for remote (automated) operation. Whatever the case may be, the media used must contain either (a) all files needed by the toolkit (itoolkit.exe, BIOS files, INI) or a batch file that points to the locations of one or more of these items.

For remote operation, the media should contain an autoexec.bat and any files to establish a network connection. Other files may also be required on the bootable media, such as ansi.sys (color display in DOS), drivers for SCSI device detection, and standard DOS files.

Manufacturing Phase

The final phase of the Integrator Toolkit usage model involves replicating selected settings onto the system during production and conducting any desired BOM operations. The four steps outlined in this phase cover the primary commands to be incorporated into the manufacturing batch files, which are created and validated in Step 4 above.

Step 6 – Flash the BIOS

Update the BIOS with the desired production BIOS release. This may be the latest available production release for the platform being used, or an earlier BIOS version that has already been used to generate a validated system image. In either case, use the toolkit to update BIOS with the **flash** command. The **force** switch can be used to update the BIOS even if the desired version is already present on the Intel desktop board. The **lang** switch may be used

to update the language pack in the BIOS.

Syntax: iToolkit FLASH -bio=<file> [-force]
 iToolkit FLASH -lang[=<path> | =<file>]

Details in Help – “Using the Toolkit | DOS-based Interface | Toolkit Reference and Formats | Command Line Options”

Step 7 – Set system date and time

Using the standard DOS commands of SET TIME and SET DATE, initialize the system date and time. Depending on the level of automation in the manufacturing environment, this can be accomplished through manual input of the correct date and time, or automated by calling a network time/date server.

Syntax: set DATE [network call]
 set TIME [network call]

Step 8 – Install BIOS configuration settings

Using the **install** command, install CMOS and SMBIOS settings on the target system. The **ini** switch can be used to specify the INI file to be used, otherwise the default filename of “itoolkit.ini” will be assumed. The **cmos** or **smbios** switches can be used to install those respective settings exclusively. The **log** switch may also be used to post command progress and success information into a specified log file.

Syntax: iToolkit FLASH -bio=<file> [-force]

Details in Help – “Using the Toolkit | DOS-based Interface | Toolkit Reference and Formats | Command Line Options”

Step 9 – Bill of Material Generation and Verification

Any BOM operations are completed at the end of the integrator toolkit workflow model. Use the **verify** command to verify that the newly manufactured system’s BOM matches that of the reference system. Use the **report** command to generate a BOM report without checking the system configuration against a reference system.

Syntax: iToolkit REPORT [-txt=<file>]
 iToolkit VERIFY [-ini=<file>] [-log[=<file>]]

Details in Help – “Using the Toolkit | DOS-based Interface | Toolkit Reference and Formats | Command Line Options”

THIS DOCUMENT AND RELATED MATERIALS AND INFORMATION ARE PROVIDED "AS IS" WITH NO WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT OF INTELLECTUAL PROPERTY RIGHTS, OR ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATION, OR SAMPLE. INTEL ASSUMES NO RESPONSIBILITY FOR ANY ERRORS CONTAINED IN THIS DOCUMENT AND HAS NO LIABILITIES OR OBLIGATIONS FOR ANY DAMAGES ARISING FROM OR IN CONNECTION WITH THE USE OF THIS DOCUMENT.

Intel Corporation may have patents or pending patent applications, trademarks, copyrights, or other intellectual property rights that relate to the presented subject matter. The furnishing of documents and other materials and information does not provide any license, express or implied, by estoppel or otherwise, to any such patents, trademarks, copyrights, or other intellectual property rights.

Intel, the Intel logo and Pentium are 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 © 2002 Intel Corporation. All Right Reserved