



Developer Guide Intel® Web 2.0 Technology Development Kit (TDK)

Version 0.6 BETA
March, 2007

Notice: This document contains information on products in the design phase of development. The information here is subject to change without notice. Do not finalize a design with this 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.

Intel products are not intended for use in medical, life saving, 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 Web 2.0 TDK 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.

Intel is a trademark or registered trademark of Intel Corporation or its subsidiaries in the United States and other countries.

Copyright © 2007, Intel Corporation. All rights reserved.

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

Principal Authorship: C.V. Vick

Contributing Authorship: Clayne Robison, Aaron Tersteeg, Jason Zhu

Filename:	Intel Web 2.0 TDK Developer Guide (0.6)
Security:	Public
Version:	v0.6
Revision Level:	37
Status:	DRAFT
Last Edit:	05/21/07 4:18 PM
Printed:	05/21/07 4:18 PM

Product License

IMPORTANT - READ BEFORE COPYING, INSTALLING OR USING.

Do not copy, install, or use the Materials provided under this license agreement ("Agreement"), until you have carefully read the following terms and conditions. By copying, installing, or otherwise using the Materials, you agree to be bound by the terms of this Agreement. If you do not agree to the terms of this Agreement, do not copy, install, or use the Materials.

End User License Agreement for the Intel Web 2.0 Technology Development Kit

1) LICENSE DEFINITIONS:

- a) "Materials" are defined as the software, documentation and other materials, including any updates and upgrade thereto, that are provided to you under this Agreement.
- b) "Sample Source" is the source code file(s) that: (i) demonstrate(s) certain limited functions that may be developed to utilize the Intel Web 2.0 Technology Development Kit product; (ii) are identified as sample source code; and (iii) are included in the technology development kit but are separate and independent from your copy of the Intel Web 2.0 TDK product.

2) LICENSE GRANT:

- a) Subject to all of the terms and conditions of this Agreement, Intel Corporation ("Intel") grants to you a non-exclusive, non-assignable, copyright license to use the Materials for your internal product development purposes only.
- b) The Materials are under development by Intel and may include software, specifications, prototypes, or other materials. Title to the Materials remains with Intel or its suppliers. You shall not mortgage, pledge or encumber the Materials in any way. You shall return all Materials, keeping no copies, upon termination or expiration of this Agreement.
- c) You may modify and redistribute the Materials or any portion of the Materials thereof, but you cannot redistribute it as the Intel Web 2.0 TDK.
- d) Subject to all of the terms and conditions of this Agreement, Intel grants to you a non-exclusive, non-assignable copyright license to modify the Sample Source, or any portions thereof, and distribute the Sample Source, or any portions thereof, as part of the product or application you developed using the Sample Source.
- e) EXCEPT AS PROVIDED HEREIN, NO OTHER LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY OTHER INTELLECTUAL PROPERTY RIGHTS IS GRANTED HEREIN.

3) OTHER LICENSE RESTRICTIONS:

- a) If you receive your first copy of the Materials electronically, and a second copy on media, or visa versa, then you may use the second copy only in accordance with your applicable license stated in this Agreement, or for backup or archival purposes. You may not provide the second copy to another user.
- b) You may NOT: (i) use or copy the Materials except as provided in this Agreement; (ii) rent or lease the Materials to any third party; (iii) assign this Agreement or transfer the Materials without the express written consent of Intel; (iv) adapt, or translate the Materials in whole or in part; (v) attempt to modify or tamper with the normal function of a license manager that regulates usage of the Materials; or (v) sublicense or transfer the Materials and derivatives thereof to any third party.

- c) This Agreement provides you with a single user license. This means you as an individual may install and use the Materials on an unlimited number of computers provided that you are the only individual using the Materials and only one copy of the Materials is in use at any one time. A separate license is required for each additional use and/or individual user in all other cases. If you are an entity, Intel grants you the right to designate one individual within your organization to have the sole right to use the Materials in the manner provided above.
- d) You shall not make any statement that your product is "certified", or that its performance is guaranteed, by Intel, by virtue of this license grant. You shall not use Intel's name or trademarks to market your product without written permission from Intel. You shall prohibit disassembly and reverse engineering of the Material and you shall not publish reviews of Materials designated as beta without written permission by Intel. You shall indemnify, hold harmless, and defend Intel and its suppliers from and against any claims or lawsuits, including attorney's fees, that arise or result from your distribution of any product.
- 4) **COPYRIGHT:** Title to the Materials and all copies thereof remain with Intel or its suppliers. The Materials are copyrighted and are protected by United States copyright laws and international treaty provisions. You will not remove any copyright notice from the Materials. You agree to prevent any unauthorized copying of the Materials. Except as expressly provided herein, Intel does not grant any express or implied right to you under Intel patents, copyrights, trademarks, or trade secret information.
- 5) **NO WARRANTY AND LIMITED REPLACEMENT:** THE 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. If the media on which the Materials are furnished are found to be defective in material or workmanship under normal use for a period of ninety (90) days from the date of receipt, Intel's entire liability and your exclusive remedy shall be the replacement of the media. This offer is void if the media defect results from accident, abuse, or misapplication.
- 6) **LIMITATION OF LIABILITY:** THE ABOVE REPLACEMENT PROVISION IS THE ONLY WARRANTY OF ANY KIND. INTEL OFFERS NO OTHER WARRANTY EITHER EXPRESS OR IMPLIED INCLUDING THOSE OF MERCHANTABILITY, NON-INFRINGEMENT OF THIRD- PARTY INTELLECTUAL PROPERTY OR FITNESS FOR A PARTICULAR PURPOSE. NEITHER INTEL NOR ITS SUPPLIERS SHALL BE LIABLE FOR ANY DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, OR OTHER LOSS) ARISING OUT OF THE USE OF OR INABILITY TO USE THE SOFTWARE, EVEN IF INTEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. BECAUSE SOME JURISDICTIONS PROHIBIT THE EXCLUSION OR LIMITATION OF LIABILITY FOR CONSEQUENTIAL OR INCIDENTAL DAMAGES, THE ABOVE LIMITATION MAY NOT APPLY TO YOU.
- 7) **UNAUTHORIZED USE:** THE MATERIALS ARE NOT DESIGNED, INTENDED, OR AUTHORIZED FOR USE IN ANY TYPE OF SYSTEM OR APPLICATION IN WHICH THE FAILURE OF THE MATERIALS COULD CREATE A SITUATION WHERE PERSONAL INJURY OR DEATH MAY OCCUR (E.G MEDICAL SYSTEMS, LIFE SUSTAINING OR LIFE SAVING SYSTEMS). Should the buyer purchase or use the Materials for any such unintended or unauthorized use, the buyer shall indemnify and hold Intel and its officers, subsidiaries and affiliates harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of product liability, personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Intel was negligent regarding the design or manufacture of the part.
- 8) **USER SUBMISSIONS:** You agree that any material, information or other communication, including all
- 9) data, images, sounds, text, and other things embodied therein, you transmit or post to an Intel website or provide to Intel under this Agreement will be considered non-confidential ("Communications"). Intel will have no confidentiality obligations with respect to the Communications. You agree that Intel and its designees will be free to copy, modify, create derivative works, publicly

display, disclose, distribute, license and sublicense through multiple tiers of distribution and licensees, incorporate and otherwise use the Communications, including derivative works thereto, for any and all commercial or non-commercial purposes.

- 10) **TERMINATION OF THIS LICENSE:** This Agreement becomes effective on the date you accept this Agreement and will continue until terminated as provided for in this Agreement. Intel may terminate this license at any time if you are in breach of any of its terms and conditions. Upon termination, you will immediately return to Intel or destroy the Materials and all copies thereof.
- 11) **U.S. GOVERNMENT RESTRICTED RIGHTS:** The Materials are provided with "RESTRICTED RIGHTS". Use, duplication or disclosure by the Government is subject to restrictions set forth in FAR52.227-14 and DFAR252.227-7013 et seq. or its successor. Use of the Materials by the Government constitutes acknowledgment of Intel's rights in them.
- 12) **APPLICABLE LAWS:** Any claim arising under or relating to this Agreement shall be governed by the internal substantive laws of the State of Delaware, without regard to principles of conflict of laws. You may not export the Materials in violation of applicable export laws.

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

Class Reference

The context awareness JavaScript library implements two types of classes: those that provide information about the platform's context, and those that provide information about the platform's devices. Device objects are a representation of the discrete "components" within the platform, e.g. CPUs, network adapters, etc. Context objects are higher level abstractions that represent environmental or situational aspects of the platform.

Helper Classes

class Event

An `Event` object wraps information about the events that are provided by the classes in the TDK and provides a well known interface between the Javascript layer and the underlying native layers. `Event` objects are not created directly by the user; only the key external methods are listed here.

Methods

Methods	Return Value	Description
<code>addObserver(EventObserver)</code>	None	After calling this method, when the event that is managed by this Event is fired, the <code>notify()</code> method on EventObserver object will be called.
<code>removeObserver(EventObserver)</code>	Number	Call this method to stop receiving notifications for this Events event.

class EventObserver

An `EventObserver` object provides the well-known interface that `Event` objects call when an event occurs. `EventObserver` objects are passed as parameters to the main `Event` methods.

Constructor

Constructor	Description
<code>EventObserver(userData)</code>	Creates a new EventObserver objects. The <code>userData</code> object will be passed to the callee in the <code>notify()</code> method.

Methods

Methods	Return Value	Description
<code>notify(userData)</code>	None	An <code>Event</code> calls this method on each subscribed <code>EventObserver</code> object whenever the event it manages occurs. An simple default implementation is provided.

class Uri

A `Uri` encapsulates URI related data.

Constructor

Constructor	Description
<code>Uri()</code>	Creates a new <code>Uri</code> object.

Properties

Property	Type	Description
<code>port</code>	Number	The port of the URI. The port in the URI http://www.intel.com:80/default.htm is 80.
<code>search</code>	String	The part of the URI to the right of the port. The Search portion of the URI http://www.intel.com:80/default.htm is "default.htm".
<code>protocol</code>	String	The protocol portion of the URI. In the URI http://www.intel.com:80/default.htm , the protocol is "http:".
<code>hostname</code>	String	The hostname portion of the URI. In the URI http://www.intel.com:80/default.htm , the hostname is "www.intel.com".

Methods

Method	Type	Description
--------	------	-------------

<code>toString()</code>	<code>String</code>	Returns a string representation of the Uri object, or in other words, the URI of the Uri object. Returns "http://www.intel.com:80/default.htm" for the URI http://www.intel.com:80/default.htm .
-------------------------	---------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Contexts

class `PowerInfo`

A `PowerInfo` object provides information about platform power conditions. An application can use this information to change its behavior depending on the amount of power available.

Constructor

Constructor	Description
<code>PowerInfo()</code>	Creates a new power information object.

Methods

Method	Type	Description
<code>getEstimatedTimeRemaining()</code>	<code>Number</code>	Returns the number of seconds remaining before the battery runs out of charge. Returns -1 if unknown, or if no battery is installed
<code>getSource()</code>	<code>String</code>	Returns the current source of power, either "Internal" or "External"
<code>getPercentRemaining()</code>	<code>Number</code>	Returns the percentage of charge remaining in the battery. Returns -1 when function call fails or if no battery is installed.
<code>getRate()</code>	<code>Number</code>	Returns the current rate of power change for the internal power sources. Discharge rate is a negative value. Charge rate is a positive value (mW/hour).

Events

Event	Arguments	Description
<code>sourceChange</code>	<code>None</code>	This event is fired when the system switches to an internal power source or vice versa. See <code>Event</code> .
<code>lowBattery</code>	<code>Threshold: Number</code>	This event is fired when user specified battery life remaining is reached. The developers can register to watch for <code><Threshold>%</code> battery life in order to take action when the specified battery life is reached.

class `ConnectivityInfo`

A `ConnectivityInfo` object provides information about the connection state of the overall system at a higher level than individual network adapters. It can be used to determine whether the platform has connectivity to a network. An application could work in cache-mode when not connected.

Constructor

Constructor	Description
<code>ConnectivityInfo()</code>	Creates a new connectivity information object.

Methods

Methods	Return Value	Description
<code>isReachable(Uri)</code>	<code>Boolean</code>	Returns true if the Uniform Resource Identifier is reachable within the default timeout and retry count. This method is limited by the browser's cross-domain scripting restrictions. If the browser's security level allows cross-domain scripting, then this method will be able to determine the availability of URIs that do not have the same hostname as the calling script. If the browser's security level does not allow cross-domain scripting, attempts to determine the availability of a URI that does not have the same hostname as the calling script will always return false .
<code>getLatency(Uri)</code>	<code>Number</code>	Returns the network latency to the host,

		where Uri is the URI of the destination. If the latency measurement times-out because the destination could not be reached, the method returns a negative number (e.g. < 0) (msec.). This method uses the <code>isReachable()</code> method and is subject to its cross-domain scripting restrictions.
<code>getDataRate(Uri)</code>	Number	Returns the instantaneous number of bytes per second that can be transferred from Uri. This method uses the <code>isReachable()</code> method and is subject to its cross-domain scripting restrictions.
<code>getIpAddresses()</code>	Array of Strings	Returns an array of IP address strings. Multiple IPAddresses may be present when the system is connected via VPN. When no active network is present, returns an empty array. Having a valid IP address does not necessarily mean that a given host is reachable. To determine if the host is reachable, the application should use isReachable() instead.
<code>isConnected()</code>	Boolean	Returns True if the system has at least one valid network interface connected to a network. This does not necessarily mean that a given host is reachable. To determine if the host is reachable, the application should use IsReachable() instead.

Events

Event	Arguments	Description
<code>connected</code>	none	Event is fired to indicate that the system has gone from a disconnected state to having at least one connection. This event is not fired if a new connection is made with a previous connection still intact.
<code>disconnected</code>	none	Event is fired to indicate that the system has lost its last network connection and is now in a disconnected state. This event is not fired if a system has multiple active connections and just one of them is lost.
<code>ipAddressTableChanged</code>	None	This event is fired whenever the system IP

		address table has changed.
<code>routeTableChanged</code>	None	This event is fired whenever the system routing table has changed.

StorageInfo

A `StorageInfo` object provides information about persistent storage and memory. This can be used by an application to check whether there is sufficient free space on disk for a large download.

Constructor

Constructor	Description
<code>StorageInfo()</code>	Creates a new storage information object.

Methods

Method	Type	Description
<code>getAvailableSpace(path)</code>	Number	Returns the available space for the logical path passed as the argument. (KB)
<code>getPersistentCapacity()</code>	Number	Returns the total capacity of all local logical disks except read-only drives, e.g. CD-drive (KB)
<code>getPersistentFreeSpace()</code>	Number	Returns the total available space for all local logical disks except read-only like CD-drives (KB)
<code>getTotalPhysicalMemory()</code>	Number	Returns the total installed physical memory (KB).
<code>getFreePhysicalMemory()</code>	Number	Returns the free physical memory (KB)

BandwidthInfo

A `BandwidthInfo` object provides information about network bandwidth. This can be used by an application to postpone the download of a large file over a low bandwidth connection. **This class is not implemented in this version of the TDK.**

Constructor

Constructor	Description
<code>BandwidthInfo()</code>	Creates a new Bandwidth information object.

Properties

Property	Type	Description
<code>getFastestProtocol()</code>	String	Returns the connected protocol with the highest transmit rate based on link speed. If a tie occurs, a wired protocol is chosen over a wireless protocol and a higher signal strength protocol is chosen over a lower signal strength protocol.
<code>getProtocolRateTx()</code>	Number	Returns the current transmit rate based on the link speed of the fastest connected protocol. (Kbps)
<code>getProtocolRateRx()</code>	Number	Returns the current receive rate based on the link speed of the fastest connected protocol. (Kbps)
<code>getSystemRateTx()</code>	Number	Returns the current aggregated transmit rate based on the link speed of all connected protocols. (Kbps)
<code>getSystemRateRx()</code>	Number	Returns the current aggregated receive rate based on the link speed of all connected protocols. (Kbps)

Methods

Method	Type	Description
<code>getProtocolRateTx(String)</code>	Number	(Kbps) Transmit rate of the protocol specified in String.
<code>getProtocolRateRx(String)</code>	Number	(Kbps) Receive rate of the protocol specified in String.

Events

Event	Type	Description
<code>bandwidthChanged</code>	none	This event is fired to indicate that the bandwidth of the system has changed.

Devices

Currently, this section contains the devices and properties we feel are relevant to applications. We can include more devices such as batteries, display adapters, etc., if there is interest.

class PlatformInfo

A `PlatformInfo` represents general platform information. **Only the `isLaptop()` method is implemented in this version of the TDK.**

Constructor

Constructor	Description
<code>PlatformInfo()</code>	Creates a new Platform object.

Methods

Method	Type	Description
<code>getManufacturer()</code>	<code>String</code>	Platform manufacturer, e.g. Quanta.
<code>getBrand()</code>	<code>String</code>	Platform brand, e.g. Dell, Lenovo. In most cases, the manufacturer and brand will be the same.
<code>getModel()</code>	<code>String</code>	Platform model. Full example: Apple's MacBook Pro Manufacturer: "Quanta Computer" Brand: "Apple" Model: "MacBook Pro"
<code>getOs()</code>	<code>String</code>	Name of the operating system, e.g. "Window XP Professional"
<code>getOsVersion()</code>	<code>String</code>	Version of the operating system, e.g. "Build 2600.xpsp.050301-1521 (Service Pack 2)"
<code>isLaptop()</code>	<code>Boolean</code>	Return true if the system is a laptop (i.e. has an internal power source).

DisplayInfo

A `DisplayInfo` object represents the primary display of the system. This can be used by an application to change its UI layout or graphics based on the resolution and orientation of the display.

Constructor

Constructor	Description
<code>DisplayInfo()</code>	Creates a new display information object.

Methods

Method	Type	Description
<code>getColorDepth()</code>	Number	Returns the number of bits per pixel used for colors on the display area.
<code>getHorizontalResolution()</code>	Number	Returns the current number of pixels in the horizontal direction (X axis) of the display.
<code>getVerticalResolution()</code>	Number	Returns the current number of pixels in the vertical direction (Y axis) of the display.
<code>getOrientation()</code>	String	Returns the current orientation of the display content, e.g “Portrait”, “Landscape”, “InvertedPortrait”, or “InvertedLandscape”

Events

Event	Arguments	Description
<code>orientationChanged</code>	None	This event is fired when the orientation of the primary display device has changed.

class Processor

A `ProcessorInfo` object represents the platform’s CPU. It can be used by applications to assess the current load on the system, or perform more computation intensive operations, e.g. higher resolution/faster frame rates, if it is a multi core system.

Constructor

Constructor	Description
<code>ProcessorInfo()</code>	Creates a new processor object.

Methods

Method	Type	Description
<code>getManufacturer()</code>	String	Returns the processor manufacturer
<code>getBrand()</code>	String	Returns a unique identifier for the

		processor
<code>getFsbFrequency()</code>	Number	Returns the frequency of Front side bus (MHz)
<code>getCpuUsage()</code>	Number	Returns the current CPU usage. (%)
<code>getLogicalCount()</code>	Number	Returns the number of logical processors currently enabled by this physical processor instance.
<code>getCpuCount()</code>	Number	Returns the number of processors (Cores, Logical, Packages, etc.) physically present on the computer system. This value does not indicate whether the operating system has enabled the cores for this application.
<code>getFeatures()</code>	Array of String	An array of the features available on the processor, e.g. hyperthreading, virtualization, SSE, etc.
<code>getCoresPerPhysical()</code>	Number	Returns the number of cores on each physical processor package. On a Dual Core system, returns 2; on a Single Core system, returns 1. This value does not indicate whether the operating system has enabled the cores for this application.
<code>getLogicalsPerCore()</code>	Number	Returns the number of logical processors per core. On a system with Hyper-threading enabled, returns at least 2. This value does not indicate whether the operating system has enabled the logical processors for this application.
<code>getEnabledCpuCount()</code>	Number	Returns the number of processors actually enabled by the Operating system for this application.
<code>getCurrentCpuFrequency()</code>	Number	Returns the current CPU Frequency. (MHz)
<code>getMaxCpuFrequency()</code>	Number	Returns the maximum CPU Frequency. (MHz)

class LinkProtocolInfo

A `LinkProtocolInfo` object represents the link protocols available on the platform. Examples include, 802.3, 802.11a/b/g, Bluetooth. **This class is not implemented in this version of the TDK.**

Constructor

Constructor	Description
<code>LinkProtocolInfo()</code>	

Methods

Method	Return Type	Description
<code>getProtocols()</code>	Array of <code>LinkProtocol</code>	Returns an array containing the link protocols available on the platform

class `LinkProtocol`

A `LinkProtocol` object represents a link protocol available on the platform.

Constructor

Constructor	Description
<code>LinkProtocol()</code>	Creates a new <code>LinkProtocol</code> object.

Methods

Method	Return Type	Description
<code>enabled()</code>	<code>Boolean</code>	Returns whether or not the protocol is available to applications. This is determined primarily by the status of the hardware to which the particular protocol applies.
<code>getIpAddress()</code>	<code>String</code>	Returns the IP address. Both IPv4 and IPv6 are supported.
<code>getMacAddress()</code>	<code>String</code>	Returns the media access control address for this network adapter.
<code>getMulticastAddresses()</code>	Array of <code>String</code>	Returns an array of IP addresses that are bound to

		the protocol as multicast addresses.
<code>getMaxLinkSpeed()</code>	Number	Returns the maximum speed of the link. (bps)
<code>getMediaPresent()</code>	Boolean	Returns true if the protocol is connected to a type of media.
<code>getState()</code>	String	Returns the state of the network adapter's connection to the network.
<code>getDataRatesRx()</code>	Array of Number	Returns an array of supported receive rates. See also (bps)
<code>getDataRatesTx()</code>	Array of Number	Returns an array of supported transmit rates. (bps)

Events

Signature	Description
<code>mediaConnected</code>	Event is fired when any protocol senses attached media and all protocols previously had no attached media.
<code>mediaDisconnected</code>	Event is fired when an instance of a protocol no longer has media attached.



Copyright © 2007 Intel Corporation. All rights reserved. BunnyPeople, Celeron, Celeron Inside, Centrino, Centrino logo, Chips, Core Inside, Dialogic, EtherExpress, ETOX, FlashFile, i386, i486, i960, iCOMP, InstantIP, Intel, Intel logo, Intel386, Intel486, Intel740, IntelDX2, IntelDX4, IntelSX2, Intel Core, Intel Inside, Intel Inside logo, Intel. Leap ahead., Intel. Leap ahead. logo, Intel NetBurst, Intel NetMerge, Intel NetStructure, Intel SingleDriver, Intel SpeedStep, Intel StrataFlash, Intel Viiv, Intel XScale, IPLink, Itanium, Itanium Inside, MCS, MMX, MMX logo,

Optimizer logo, OverDrive, Paragon, PDCharm, Pentium, Pentium II Xeon, Pentium III Xeon, Performance at Your Command, Pentium Inside, skool, Sound Mark, The Computer Inside., The Journey Inside, VTune, Xeon, Xeon Inside and Xircom 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.