ESTTA Tracking number: **ESTTA683623**
Filing date: **07/14/2015**

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE TRADEMARK TRIAL AND APPEAL BOARD

Proceeding	77844736
Applicant	Apple Inc.
Applied for Mark	OPENCL
Correspondence Address	GLENN A GUNDERSEN DECHERT LLP CIRA CENTRE, 2929 ARCH STREET PHILADELPHIA, PA 19104-2183 UNITED STATES trademarks@dechert.com, glenn.gunderson@dechert.com, hal.borden@dechert.com, jacob.bishop@dechert.com, trade-marks@dechert.com
Submission	Appeal Brief
Attachments	OPENCL -- Appeal Brief.pdf(140260 bytes) OPENCL -- Appeal Brief (Exhibits A-E).pdf(2484910 bytes) OPENCL -- Appeal Brief (Exhibits F-J).pdf(4199529 bytes)
Filer's Name	Jacob Bishop
Filer's e-mail	trademarks@dechert.com, glenn.gundersen@dechert.com, erin.reimer@dechert.com
Signature	/Jacob Bishop/
Date	07/14/2015

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE TRADEMARK TRIAL AND APPEAL BOARD

In re Application of		:
Apple Inc.		:
		:
Mark:	OPENCL and Design	:
Serial No.:	77/844,736	:
Filing Date:	October 8, 2009	:

APPLICANT'S APPEAL BRIEF

Index of Cited Cases

Bongrain International (American) Corporation v. Delice de France, Inc.,
811 F.2d 1479, 1 U.S.P.Q.2d 1775 (CAFC 1987) 7

INTRODUCTION

Applicant Apple Inc. ("Apple") seeks to register the mark OPENCL and Design (as presented below) for "application programming interface computer software for use in developing applications for execution on central processing units (CPU) or graphic processor units (GPU), sold as an integral component of computer operating software" in International Class 9 (the "OPENCL Logo").

Apple uses its OPENCL mark in connection with a computing open standard created by Apple and licensed to others in the computer industry, as well as the software used to develop applications consistent with that standard, which is sold as part of OS X, Apple's computer operating system software. It is seeking to register the mark for the software.

Apple already owns a U.S. registration of OPENCL in standard characters for the same goods covered by the present application. Apple established use of the OPENCL mark for these goods by showing the mark used in connection with Apple's software. Apple now seeks to register its OPENCL mark paired with the distinctive "speedometer" logo displayed above.

The prior Examining Attorney initially refused registration of both the OPENCL word mark and the OPENCL Logo on the grounds that OPENCL is merely descriptive for Apple's goods, claiming that, because OPENCL identifies an open standard and a computing language, the term is incapable of functioning as an indicator of source for Apple's software. Apple appealed the Examining Attorney's decision, and the Trademark Trial and Appeal Board

reversed the Examining Attorney's descriptiveness refusal. Both applications were subsequently published without third party opposition; the OPENCL application proceeded to registration, while the present application for the OPENCL Logo was allowed.

In December 2013, Apple submitted a Statement of Use for the present application, and included the following printout from Apple's website as a specimen of use for the OPENCL Logo:

The prior Examining Attorney again refused registration of the OPENCL Logo, on the grounds that (1) the term OPENCL cannot function as a trademark for Apple's computer software (since it identifies an open standard and a computing language), (2) Apple's specimen of use does not show use of the OPENCL mark in connection with the goods covered by its

application, and (3) the goods covered by Apple's application are not "goods in trade." Apple has appealed the prior Examining Attorney's refusal.

This appeal presents the extraordinary situation in which the same application has been refused twice, and the refusal appealed to the Board twice, on essentially the same grounds. In the first refusal, the Examining Attorney contended that OPENCL was incapable of functioning as an indicator of source for Apple's software because it identifies an open standard and a computing language. The new refusal makes essentially the same argument, but takes the form of a refusal of Apple's specimen of use. Between the first appeal and second refusal, this application was published for opposition, without any opposition or extension of time to oppose. Nevertheless, the Examining Attorney continues to refuse registration, notwithstanding the fact that no one in the computer industry objected to Apple's registration of OPENCL as a word mark or in logo form, and that several leading companies in the industry have in fact provided affidavits stating that the industry recognizes OPENCL as Apple's trademark.

ARGUMENT

1. OPENCL FUNCTIONS AS A TRADEMARK.

A. Apple already owns a federal registration of OPENCL for the same goods.

As mentioned above, Apple already owns U.S. Registration No. 4,268,483 of the mark OPENCL for the very same goods that are covered by the present application. The registration issued after the Board overturned the Examining Attorney's refusal to register the mark. The OPENCL application was then published for opposition, and was unopposed. By virtue of its existing registration, Apple already enjoys a legal presumption that it has the exclusive nationwide right to use the mark OPENCL for its goods, not limited to any particular logo or

font. The present application merely seeks to confirm Apple's right to use OPENCL in the particular distinctive logo shown in the drawing.

B. Apple has provided substantial evidence showing that the computer industry views OPENCL as Apple's trademark.

At earlier stages in the prosecution of this application and the companion OPENCL application, Apple submitted affidavits from the leading companies in the computer industry stating that OPENCL is an Apple trademark— specifically, Apple submitted affidavits from representatives of Khronos, which licenses the OPENCL mark on Apple's behalf, and NVIDIA Corporation and Advanced Micro Devices, Inc. (AMD), two companies that have adopted the OPENCL standard. Copies of these affidavits are once again submitted as **Exhibit A** and **Exhibit B**. The industry's recognition of OPENCL as an Apple trademark is also evident in a Google search of OPENCL, where each of the top entries refers to Apple's ownership of the mark:

- The top hit is the Wikipedia entry for OPENCL – which the Examining Attorney has submitted as evidence – and which states that "OpenCL was initially developed by Apple, which holds trademark rights."
- The websites of Kronos, NVIDIA, and AMD all refer to OPENCL as an Apple trademark.
- Intel's website displays OPENCL with a "TM" legend and states that "OpenCL and the OpenCL logo are trademarks of Apple Inc."

See **Exhibit C**.

The Examining Attorney has not addressed this evidence, and instead attempted to find his own evidence to show that OPENCL is not a trademark. However, he has not produced any

authoritative evidence to support the claim that Apple is not using OPENCL as a trademark, and in refusing registration, the Examining Attorney has ignored evidence in the record—coming directly from those in the industry who are the most knowledgeable—that refutes his position that OPENCL does not function as Apple's trademark.

In a series of opinions in the 1980s and 1990s that are now cited in the *Trademark Manual of Examining Procedure*, the Court of Appeals for the Federal Circuit overturned several decisions in which the Board found a likelihood of confusion between two marks, even though the two trademark owners had agreed in writing that no confusion existed. In *Bongrain International (American) Corporation v. Delice de France, Inc.*, 811 F.2d 1479, 1 U.S.P.Q.2d 1775 (CAFC 1987), for example, the Court stated as follows:

> "We have often said, in trademark cases involving agreements reflecting parties' views on the likelihood of confusion in the marketplace, that they are in a much better position to know the real life situation than bureaucrats or judges."

The Court noted that "[t]he primary purpose of the Trademark Act of 1946 is to give Federal procedural augmentation to the common law rights of trademark owners – which is to say legitimate users of trademarks."

The situation in this application is analogous. Apple has shown that those in the industry who use Apple's OPENCL software consider OPENCL to be Apple's trademark. There is no doubt that Apple is using OPENCL as a trademark for computer software, and there is no doubt that the specimen of use leads to a web page where developers can download Apple's software.

Apple already owns a registration of OPENCL that carries a legal presumption that OPENCL is Apple's trademark for the computer software identified in that application. That presumption would not change if Apple chose to abandon the present application. OPENCL

would still continue to be Apple's registered trademark, but the distinctive OPENCL logo would not appear on the trademark register.

To continue to refuse registration of the OPENCL logotype would frustrate the purpose of the Trademark Act, as enunciated in *Bongrain*. Apple has been using the OPENCL logo since 2009, and the industry explicitly acknowledges it to be Apple's mark. As the Court said in *Bongrain* and subsequent decisions, those in the industry "are in a much better position to know the real life situation than bureaucrats or judges."

C. The name of an open standard or computing language can also function as an indicator of source for computer software.

The Examining Attorney asserts the mistaken notion that the name of a technical computing standard and computing language cannot function as a trademark.

The Examining Attorney's initial basis for the definition of "standard" was a website called "Wigglebits", which was neither authoritative nor relevant. The Examining Attorney based the final refusal on just one piece of evidence – a definition of "industry standard" found on the website of the National Institute of Standards and Technology (NIST). While an organization like NIST has more credibility than Wigglebits, it has nothing to do with technical computing standards. Its definition applies to its own activities, but is not at all relevant to Apple's use of the OPENCL mark.

NIST defines an "industry standard" as "a voluntary, industry-developed document that establishes requirements for products, practices or operations." This definition is plainly inapplicable in the current situation, because OPENCL is not "a voluntary, industry-developed document" (OPENCL was initially developed by Apple), nor does it "establish requirements for

products, practices or operations" (OPENCL does not "establish requirements" for producing a product).

It's clear from NIST's own mission statement that it has nothing to do with computing standards such as OPENCL. NIST is a "physical science" laboratory – i.e., dealing with physical objects – established to create a "measurement infrastructure" used today to measure objects as small as tiny nanoscale devices and as large as skyscrapers. The organization's website states its mission as follows:

> "NIST is one of the nation's oldest physical science laboratories. Congress established the agency to remove a major handicap to U.S. industrial competitiveness at the time – a second-rate measurement infrastructure that lagged behind the capabilities of the United Kingdom, Germany, and other economic rivals. Today, NIST measurements support the smallest of technologies – nanoscale devices so tiny that tens of thousands can fit on the end of a single human hair – to the largest and most complex of human-made creations, from earthquake-resistant skyscrapers to wide-body jetliners to global communication networks."

See **Exhibit D**. Thus, the refusal is based on a definition of "standard" that has no relevance to technical computing standards in the computer industry.

In contrast, the Wikipedia entry for "open standard" for computing supports Apple's position. It demonstrates that there is no single definition of the term, and in fact cites twenty different possible definitions:

> "There is no single definition and interpretations vary with usage. The terms 'open' and 'standard' have a wide range of meanings associated with their usage. There are a number of different definitions of open standards which emphasize different aspects of openness, including … the ownership of rights in the standard… Many specifications that are sometimes referred to as standards are proprietary and only available under restrictive contract terms (if they can be obtained at all) from the organization that owns the copyright on the specification." (http://en.wikipedia.org/wiki/Open_standard#Programming_languages)

See **Exhibit E**. This entry confirms that, as in Apple's case, a standard can be proprietary, and give rise to trademark rights, and refutes the Examining Attorney's rationale for refusing registration.

As Apple has previously demonstrated, OPENCL signifies both (1) a computing open standard created by Apple and licensed to others in the computer industry, and (2) the software used to develop applications consistent with that standard. Apple is entitled to register the OPENCL logo for the latter.

Those unfamiliar with the industry may misconstrue the phrase "open standard," mistakenly assuming that it means that the technology is not proprietary, or that the name of the standard doesn't function as an indicator of origin. However, that's not the case. In order to clarify the technical and specialized nature of open standards, Apple has previously submitted affidavits and evidence clarifying the nature of open standards generally, and its use of the OPENCL mark specifically. This evidence demonstrates that the industry recognizes OPENCL as an indicator of origin, with Apple as the source.

Apple operates an extensive software development program in which it provides access to technology that allows developers to create software applications to run on Apple devices, thereby making Apple products even more attractive to software users. Consumers are familiar with the apps that developers create for use on Apple's iPhone and iPad devices; OPENCL represents a software creation framework that will be unfamiliar to the average consumer, but is well-known in the industry. Industry leaders like Apple create such frameworks, and provide access to those frameworks to allow others to build their software applications using the tools that they have created.

OPENCL is a technical framework that Apple created in order to allow computer programmers to write software for devices with multiple processors. In order to popularize this framework, Apple began working with The Khronos Group, an industry consortium of leading technology companies, to develop this framework into an open standard for use in the computing industry. Apple granted Khronos a license to use the mark OPENCL in connection with the ongoing development of the technical specifications for the standard. These specifications are not determined by general consensus or common usage, but solely by Khronos. Members of Khronos are licensed to use the mark OPENCL in connection with implementations of the standard that conform to the specifications, as determined by Khronos. Apple itself uses OPENCL as a mark for its own implementation of the standard—an application program interface (API) software feature of the Mac OS X operating system.

2. THE SPECIMEN OF USE DEMONSTRATES USE OF THE MARK IN COMMERCE FOR THE GOODS COVERED BY THE APPLICATION.

Apple's specimen of use displays the OPENCL Logo above eight links to "References, Guides, and Sample Code," which developers can follow to download an implementation of Apple's application programming interface computer software, and to access a "Programming Guide" for using the application programming interface computer software. See **Exhibit F**. The Examining Attorney states that "it is unclear how consumers would identify 'sample code'…as "application programming [interface] computer software." However, a developer will clearly understand that. As shown in the attached **Exhibit G**, an application programming interface is a type of software consisting of a "set of routines, protocols, and tools" that software developers use to create software products. Application programming interfaces can come in the form of "a

library that includes specifications for routines, data structures, object classes, and variables," and are often times closely associated with a programming language. Moreover, their specifications can take the form of an "industry standard." Thus, contrary to the assertions in the final refusal, Apple's specimen—which displays the OPENCL Logo in connection with a range of programming tools and sample code written in the OPENCL programming language—clearly shows use of the OPENCL Logo in connection with the goods covered by the application.

The Trademark Office has issued all of the following federal registrations that cover "application programming interface computer software":

- MAMDA (Reg. No. 4288465) (NYSE Technologies, Inc.)
- PXML (Reg. No. 3677861) (Printronix, Inc.)
- DynamicOSI (Reg. No. 4280967) (Monfox LLC)
- MxLWARE (Reg. No. 3854657) (MaxLinear, Inc.)

See **Exhibit H**. Tellingly, each of these entities submitted specimens of use comprised of user guides or developer guides that clearly identify that their "application programming interface computer software" is, like Apple's OPENCL software, comprised of tools and computer languages used by software developers to create software applications. See **Exhibit I**. Indeed, these specimens of use are conceptually similar to Apple's OPENCL "Programming Guide" referenced in Exhibit F above. These examples highlight the fact that the relevant class of consumers—computer programmers—will not see an appreciable difference between the terms "code" and "application programming interface computer software," but will instead understand that Apple uses its mark in connection with the software covered by its application.

The Board has already once concluded that the following specimen was sufficient to show use of the OPENCL word mark for "application programming interface computer

software":

The present case is not substantively different from the OPENCL word mark application. The differences between the two specimens are immaterial. The only difference is that, in the present case, Apple has submitted the page from its website that contains the link to the specimen used for the OPENCL word mark application. See **Exhibit J**. As a practical matter, developers will still view Apple's use of the OPENCL Logo contemporaneously with their download of the implementation of Apple's OPENCL application programming interface computer software and the OPENCL "Programming Guide."

CONCLUSION

For the reasons presented above, Apple respectfully submits that the prior Examining Attorney's conclusion that the OPENCL Logo does not function as a trademark is unfounded, and that the specimen submitted in its statement of use meets the Trademark Office's requirements for specimens for software products. Apple therefore respectfully requests that Application Serial No. 77/844,736 be approved for publication.

Respectfully submitted,

/Jacob Bishop/

Glenn A. Gundersen
Jacob Bishop
Dechert LLP
Cira Centre
2929 Arch Street
Philadelphia, PA 19104-2808
Telephone: 215.994.2183
Fax: 215.655.2183
trademarks@dechert.com

Attorney for Applicant
APPLE INC.

Exhibit A

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of Apple Inc.	:	Trademark Attorney: Michael Webster
	:	Law Office: 102
Mark: OPENCL & Design	:	
Serial No. 77/844,718	:	**DECLARATION OF** **NEIL TREVETT**

---

1. I, Neil Trevett, am the President of the Khronos Group, a not-for-profit industry consortium that manages open standards for the authoring and acceleration of parallel computing, graphic, and dynamic media on a wide variety of computing platforms and devices. I am also Vice President of Mobile Content at NVIDIA Corporation, one of the world's leading developers of digital media processors. All information provided within this affidavit is personally known to me or is information that has been provided to me that I believe to be true.

2. I have worked professionally in the computer industry for more than 20 years. From 1997 to 2005, I served as President of the Web3D Consortium, which organizes industry-leading companies dedicated to creating open standards for communicating real-time in 3D on the Internet. I was elected President of the Khronos Group in 2001, and I joined NVIDIA as vice president in 2005.

3. Khronos was founded in 2000 by a number of media-centric companies, including NVIDIA, ATI Technologies, Discreet, Evans & Sutherland, Intel Corporation, Silicon Graphics, and Sun Microsystems. Today, Khronos Group has roughly 100 member companies.

4. Khronos Group manages a number of different computing standards that were originally developed by other parties. One of these is the OPENCL standard, which is based on technology developed by Apple Inc. and is used as a framework for writing programs that execute across heterogeneous devices consisting of CPU's, GPU's, and other processors.

5. OPENCL is not generic or an industry term of art. It is also not merely the name of a computer language. OPENCL refers to the standard originally developed by Apple, which Apple has contributed to the Khronos Group, together with a trademark license for OPENCL from Apple to Khronos so that Khronos Group can manage the ongoing development of the technical specifications and administer a conformance program for the standard. The members of Khronos — namely, companies such as

NVIDIA, IBM, AMD, and Intel — are licensed to use the mark OPENCL in connection with their own conforming implementations of the standard through the conformance program.

6. The specifications of the OPENCL standard are not determined by general consensus or common usage; they are determined by working groups within the Khronos Group consisting solely of Khronos members. Developers who see the mark OPENCL in connection with a member's implementation of the standard know that the implementation meets the specifications promulgated by Khronos Group and have passed the criteria defined in Khronos' conformance program. Our trademark usage guidelines (attached as Exhibit A) specify how our members may use the names of the standards we manage, including the mark OPENCL, which we have licensed from Apple.

7. Based on my experience with industry consortia in the software development field, it is clear that the name of an open standard can also be the name of a programming language, but that the industry would still recognize that name as a trademark. Developers associate the name of an open standard with the organization managing and evolving the standard, and developers use the mark as an indication of conformance to the criteria defined by that organization. The names of the standards that Khronos Group manages are trademarks, and our members that implement the standards are using the marks under license.

8. Apple uses OPENCL as a mark for a particular implementation of the OPENCL standard and programming language—namely, an application program interface (API) computer software feature of the Mac OS X operating system. This feature meets the specifications of the OPENCL standard managed by Khronos under license from Apple.

Having been warned that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 10001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application or any registration resulting therefrom, I further declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true.

Neil Trevett

Exhibit A

Khronos Trademark Guidelines
April 2010

1. Background

Khronos owns, or has licensed, trademarks used to identify its activities and specifications – for example the OpenGL® mark. The trademarks used by Khronos are collectively referred to as the *Khronos Marks*. Khronos maintains strict control over usage of the Khronos Marks to protect the integrity of its specifications. This document defines who may use the Khronos Marks, and under what terms and conditions.

Khronos Marks are protected in both text and logo form. For example, the text string "OpenGL" is a trademark - as is the OpenGL logo. In addition, Khronos may make available Certification Logos available for use on fully conformant products.

To assure that implementations of Khronos specifications meet a consistent standard of the highest quality, the Khronos Group maintains Adopter Programs that provide access to conformance tests for verification by Khronos that a product or implementation conforms to the specification. Any party who wishes to market and distribute an implementation that is identified by use of a Khronos Mark (in text or logo form) must execute the Khronos Adopters Agreement *http://www.khronos.org/files/adopters_agreement.pdf* and successfully pass the implementation through the conformance testing procedure as defined in the Khronos Conformance Process Document *http://www.khronos.org/files/conformance_procedures.pdf*.

A *Khronos Member* has a current Promoter or Contributor membership of the Khronos Group, and an *Adopter* has executed a Khronos Adopters Agreement for a specific specification and has paid any associated Adopters fees. You do not have to be a Khronos Member to be an Adopter – and vice versa.

Use of Marks NOT RELATED to a Product or Platform	Use of Marks in association with a Product or Platform – MUST BE BY AN ADOPTER (paid and signed)				
	Use of Marks in association with a Product Implementation				Use of Marks in Platform Definitions
	Spec is Provisional or tests are not published for API	Product is NOT submitted to Conformance Process	Product Passes Published Conformance Tests – BUT Submission is in 30 Day Review	Product Passes Published Conformance Tests – AND Submission is Complete	Platform definition must mandate Khronos Conformance for referenced specifications
Members Permission granted through Membership Agreement	Use text mark only – NOT Logo	Use text mark only – NOT Logo	May use text mark AND Logo	May use text mark AND Logo	May use text mark AND Logo
Non-Members Must obtain written permission from Khronos	Make agreed statement that specification is provisional and implementation may change when finalized	Make agreed statement that Product is not yet conformant – and provide link to conformance site	Make agreed statement that Product is in Conformance review – and provide link to conformance site	May also optionally use Certification Logo	
Example: Promoting publications, courses					
Khronos published Trademark Usage Guidelines MUST be followed (includes guidelines from trademark licensors)					

Figure 1. Summary of Trademark Guidelines for Khronos Marks

2. Use of Khronos Marks NOT Related to a Product or Platform

Companies or individuals that are not members of Khronos and who wish to use a Khronos Mark for promotional or educational purposes are welcome to apply for a Khronos Non-Member Trademark License at *www.khronos.org/trademarks*. Grant of the Non-Member Trademark License is at Khronos' sole discretion and specifically excludes any usage of a Khronos Mark (in text or logo form) in association with a specific implementation of a Khronos Specification.

Khronos members may use Khronos Marks for promotional and educational purposes according to the terms and conditions in the Khronos Membership Agreement. Members that are not Adopters of a specification are excluded from using a Khronos Mark (in text or logo form) in association with a specific implementation of that Khronos Specification.

Examples:

- a publisher who wishes to use the OpenCL logo on the cover of a book must apply to Khronos for a Non-Member Trademark License – which grants permission for that specific trademark usage;

- a Khronos member that is not an Adopter of OpenGL may use the OpenGL logo on a web page that is promoting high performance graphics – as long as no specific implementation of OpenGL is referenced;

- a Khronos member that is NOT an Adopter of OpenVG may NOT use the OpenVG name OR logo in association with a specific hardware or software implementation of the OpenVG specification.

3. Use of Khronos Marks in Association with a Product Implementation

ANY company using a Khronos Mark (in text or logo form) in association with an implementation of a Khronos specification MUST be a fully paid and signed up Adopter of that specification.

If a product has not fully passed the Khronos Conformance Process an Adopter may NOT use wording associated with that product such as "conformant", "compliant", "complies with", "full implementation" or any other term that implies in any way that the implementation is officially compliant.

3.1 Provisional specification or conformance tests not available

An Adopter with an implementation of a provisional specification, or an implementation of a specification that does not have available conformance tests, may use the relevant Khronos Mark in association with the implementation with the following restrictions:
a) only in text and *not* logo form;
b) any materials using the Khronos Mark must state as appropriate:

"Product is based on a provisional Khronos Specification, which may change before final release. Current specification status can be found at www.khronos.org/specname"
Or
"Product is based on a published Khronos Specification, and is expected to pass the Khronos Conformance Testing Process when available. Current conformance status can be found at www.khronos.org/conformance."

3.2 Products before submission to conformance process

An Adopter with an implementation of a published specification that has not yet been submitted to the Khronos Conformance Process may use the relevant Khronos Mark in association with the implementation with the following restrictions:
a) only in text and *not* logo form;
b) any materials using the Khronos Mark must state:

"Product is based on a published Khronos Specification, and is expected to pass the Khronos Conformance Testing Process. Current conformance status can be found at www.khronos.org/conformance."

3.3 Products during Submission Review Period

An Adopter with an implementation of a published specification that has been submitted to the Khronos Conformance Process and is in the Conformance Review Period, including any restarts of the Review Period, may use the relevant Khronos Mark in text *and* logo form in association with the implementation with the following restriction:
a) any materials using the Khronos Mark must state:

"Product is based on a published Khronos Specification, and is expected to pass the Khronos Conformance Testing Process. Current conformance status can be found at www.khronos.org/conformance."

3.4 Conformant Products

An Adopter that has an implementation of a published specification that has been submitted to and passed the Khronos Conformance Process may use the relevant Khronos Mark in text, logo and certification mark form in association with the implementation, and Adopters may state that their products are conformant or compliant with the appropriate specifications with statements similar to:

"Product is conformant with [Khronos Specification]"

"Product is compliant with [Khronos Specification]"

"Product is complies with [Khronos Specification]"

"Product is a full implementation with [Khronos Specification]".

or use other terms as appropriate that communicates that the implementation is officially compliant.

4. Use of Khronos Marks in Platform Definitions

ANY company using a Khronos Mark (in text or logo form) within a platform or operating system definition that references a Khronos specification MUST be a fully paid and signed up Adopter of that specification.

Adopters may use a Khronos Mark in an operating system specification or platform definition, in text or logo form, provided that Khronos Conformance for products implementing the referenced specifications is mandated as part of the conformance procedure for that platform.

5. Trademark Usage

Any party that is granted use of Khronos Marks must follow Khronos Trademark Usage Guidelines at *www.khronos.org/trademarks*.

In addition:

Apple has licensed the OpenCL trademark to Khronos, and any use of the OpenCL mark must follow both Apple's Guidelines for Using Apple Trademarks found at http://www.apple.com/legal/trademark/guidelinesfor3rdparties.html, and any additional OpenCL Logo Guidelines posted by Apple.

SGI has licensed the OpenGL, OpenML, OpenGL ES and OpenGL SC trademarks to Khronos, and any use of the SGI Marks must follow SGI's guidelines for using SGI trademarks found at http://www.sgi.com/company_info/trademarks.

Exhibit B

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of Apple Inc.	:	Trademark Attorney: Michael Webster
	:	Law Office: 102
Mark: OPENCL	:	
Serial No. 77/616247	:	**DECLARATION OF** **WILLIAM W. LICEA-KANE**

---

1. I, William W. Licea-Kane, am a Principal Member Technical Staff of Advanced Micro Devices, Inc. (AMD), one of the world's leading semiconductor design developers, with particular emphasis on graphics and computing technologies used in computers, computer servers, gaming devices and embedded technologies. All information provided within this affidavit is personally known to me or is information that has been provided to me that I believe to be true.

2. I have worked professionally in the computer industry for more than thirty years. Since 2001, I have served as a member of technical staff of the graphics product group (formerly ATI, Inc, acquired by AMD in 2006) within AMD, specifying Application Programming Interfaces (API), programming languages and driver software for AMD's graphics and computing technologies.

3. In my capacity as Principal Member Technical Staff of AMD, I have been involved in the activities of Khronos Group, an organization founded in 2000 by a number of media-centric companies. AMD is a promoter member of Khronos Group.

4. Khronos Group manages a number of different computing standards that were developed by other parties. One of these is the OPENCL standard, which is based on technology developed by Apple Inc. and is used as a framework for writing programs that execute across heterogeneous devices consisting of CPU's, GPU's, and other processors.

5. OPENCL is not generic or an industry term of art. It is also not merely the name of a computer language. OPENCL refers to the standard developed by Apple, which Apple has licensed to Khronos Group, so that Khronos Group can manage the ongoing development of the technical specifications for the standard. Members of Khronos Group — including AMD — use the mark OPENCL under license in connection with their own implementations of the standard.

6. The specifications of the OPENCL standard are not determined by general consensus or common usage; they are determined solely by Khronos Group. As such, it is my opinion that developers who see the mark OPENCL in connection with a member's implementation of the standard know that the implementation meets the specifications promulgated by Khronos Group. The Khronos Group trademark usage guidelines (attached as Exhibit A) specify how Khronos Group members, including AMD, may use the names of the standards managed by Khronos Group, including the mark OPENCL.

7. Based on my experience with industry consortia in the software development field, it is clear that the name of an open standard can be the name of a programming language, and that the industry would recognize that name as a trademark. Developers recognize the name of an open standard as identifying the innovator behind the standard, and developers use the mark under a license from that source. The names of the standards that Khronos Group manages are trademarks, and AMD and other members that implement the standards are using the marks under license.

8. AMD uses OPENCL as a mark to identify a particular implementation of the OPENCL standard and programming language—namely, an application program interface (API) computer software feature. This feature meets the specifications of the OPENCL standard managed by Khronos under license from Apple.

Having been warned that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application or any registration resulting therefrom, I further declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true.

William W. Licea-Kane
Date : August 24, 2011

2011-08-24

Exhibit C

Google opencl

Web News Books Videos Images More Search tools

About 1,100,000 results (0.22 seconds)

Open Computing Language (**OpenCL**) is a framework for writing programs that execute across heterogeneous platforms consisting of central processing units (CPUs), graphics processing units (GPUs), digital signal processors (DSPs), field-programmable gate arrays (FPGAs) and other processors.

OpenCL - Wikipedia, the free encyclopedia
en.wikipedia.org/wiki/**OpenCL** Wikipedia

More about OpenCL

Feedback

OpenCL - Khronos Group
https://www.khronos.org/**opencl**/ Khronos Group
Vulkan · **OpenCL** · OpenGL; OpenGL ES; WebGL; WebCL; COLLADA; glTF; EGL; OpenSL ES; OpenMAX; SPIR; SYCL; StreamInput; OpenVX; OpenKCam ...
Khronos OpenCL Registry - Reference Cards - Online Man pages - OpenGL

OpenCL - Wikipedia, the free encyclopedia
en.wikipedia.org/wiki/**OpenCL** Wikipedia
Open Computing Language (**OpenCL**) is a framework for writing programs that execute across heterogeneous platforms consisting of central processing units (CPUs), graphics processing units (GPUs), digital signal processors (DSPs), field-programmable gate arrays (FPGAs) and other processors.
Overview - History - Implementation - Timeline of vendor implementations

OpenCL - NVIDIA Developer
https://developer.nvidia.com/**opencl** Nvidia
OpenCL™ (Open Computing Language) is a low-level API for heterogeneous computing that runs on CUDA-powered GPUs. Using the **OpenCL** API, developers ...

OpenCL™ Technology | Intel® Developer Zone
https://software.intel.com/en-us/intel-**opencl** Intel
OpenCL support is now integrated into Intel's software development suites: Intel® Integrated Native Developer Experience (Intel® INDE) and Intel® Media ...

The AMD OpenCL™ Zone - development for AMD
developer.amd.com › Tools & SDKs Advanced Micro Devices
Accelerate your applications on CPUs, GPUs & APUs. Find the tools for a complete acceleration platform in AMD's **OpenCL**™ Zone.

Developer Zone

Development > Tools > Resources >

OpenCL™ Technology

- OpenCL support is now integrated into Intel's software development suites: Intel® Integrated Native Developer Experience (Intel® INDE) and Intel® Media Server Studio
- Intel® SDK for OpenCL™ applications features are now under the new name of OpenCL™ Code Builder.

Integrated OpenCL Development Experience

Intel® SDK for OpenCL™ Applications features are now integrated into various development tools under the new name of OpenCL™ Code Builder:

Tailored to your development environment.

- For mobile and PC development, the OpenCL™ Code Builder is now integrated into Intel® Integrated Native Developer Experience (Intel® INDE).
- For media server usages, the OpenCL™ Code Builder is now integrated into Intel® Media Server Studio, Including the new commercial OpenCL Linux driver for Intel Graphics.
- For other usages, standalone Linux* version of the OpenCL™ Code Builder is also available here

OpenCL and the OpenCL logo are trademarks of Apple Inc. used by permission by Khronos.

Copyright ©Intel Corporation. All rights reserved. Intel Corporation, 2200 Mission College Blvd., Santa Clara, C

OpenCL

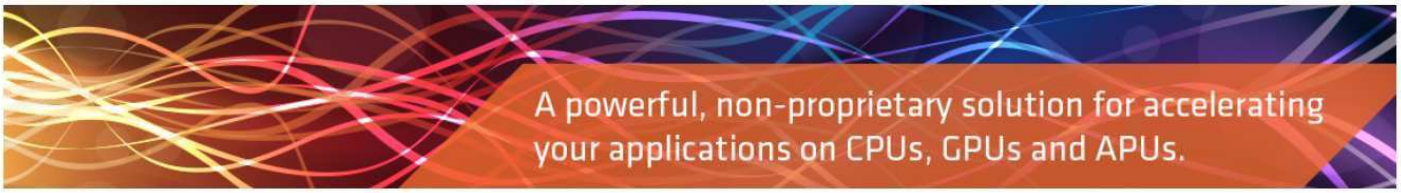
OpenCL™ (Open Computing Language) is a low-level API for heterogeneous computing that runs on CUDA-powered GPUs. Using the OpenCL API, developers can launch compute kernels written using a limited subset of the C programming language on a GPU.

OpenCL support is included in the latest NVIDIA GPU drivers, available at www.nvidia.com/drivers

In addition to OpenCL, NVIDIA supports a variety of GPU-accelerated libraries and high-level programming solutions that enable developers to get started quickly with GPU Computing.

OpenCL is a trademark of Apple Inc., used under license by Khronos.

OpenCL™ Zone – Accelerate Your Applications



OpenCL™ is the first truly open and royalty-free programming standard for general-purpose computations on heterogeneous systems. Open Computing Language allows programmers to preserve their expensive source code investment and easily target multi-core CPUs, GPUs, and the new APUs. Developed in an open standards committee with representatives from major industry vendors, OpenCL™ gives users what they have been demanding: a cross-vendor, non-proprietary solution for accelerating their applications on CPU, GPUs and APUs. AMD, an early supporter of OpenCL™ and leading innovator and provider of high-performance CPUs and GPUs, is uniquely positioned in this industry to offer a complete acceleration platform for OpenCL™. Developer and technology partners have created several applications, libraries and technology demonstrations taking advantage of AMD Accelerated Parallel Processing acceleration, also abbreviated as "APP." Use the resources below to get started with OpenCL™ today.

Tools & SDKs

AMD OpenCL™ APP SDK – execute heterogeneously to accelerate many applications beyond just graphics

CodeXL – A comprehensive tool suite that includes powerful debugging, profiling, and static OpenCL™ kernel analysis capabilities

Legacy AMD OpenCL™ Tools — These tools have lived a long and fruitful life. Now discontinued but provided here without warranty as a convenience

Libraries

ACML — ACML includes optimized and threaded linear algebra math libraries for use in for high performance computing, scientific, engineering, and related compute-intensive applications for use on AMD processors. Includes BLAS, LAPACK, FFT, and RNG routines.

clMagma — OpenCL™ port of the MAGMA project, a linear algebra library similar to LAPACK but for Multicore+GPU systems

clMath — OpenCL™ software libraries containing FFT and BLAS functionalities

Bolt — An STL compatible library for creating accelerated data parallel applications

GPUPerfAPI — AMD's library for accessing GPU performance counters on ATI Radeon graphics cards

PARALUTION is a sparse linear algebra library with OpenCL™ support that has a rich collection of iterative solvers and preconditioners. The code can be compiled under Linux/Unix, Mac OS and Windows

Resources

Getting Started with OpenCL™

Programming in OpenCL™

Training — Check out these AMD partners who offer a variety of OpenCL™ training options customized to fit your needs

OpenCL™ Articles and Papers

OpenCL™ Technical Overview Video Series

OpenCL™ On-Demand Webinar Series

Support

AMD OpenCL™ Developer Forums

AMD Developer Forums Home Page

OpenCL and the OpenCL logo are trademarks of Apple Inc. used by permission by Khronos

Exhibit D

About NIST

Welcome to the National Institute of Standards and Technology's web site. Founded in 1901 and now part of the U.S. Department of Commerce, NIST is one of the nation's oldest physical science laboratories. Congress established the agency to remove a major handicap to U.S. industrial competitiveness at the time—a second-rate measurement infrastructure that lagged behind the capabilities of the United Kingdom, Germany, and other economic rivals. Today, NIST measurements support the smallest of technologies—nanoscale devices so tiny that tens of thousands can fit on the end of a single human hair—to the largest and most complex of human-made creations, from earthquake-resistant skyscrapers to wide-body jetliners to global communication networks. We invite you to explore our web site to learn about our current projects, to find out how you can work with us, or to make use of our products and services.

This section of our web site contains basic information about NIST. If you can't find the information you need, contact Public Inquiries at (301) 975-NIST or inquiries@nist.gov.

- General information -- links to program information, speeches, testimony, frequently asked questions.
- The Story of NIST -- online timeline of NIST's history from 1901 to the present
- Budget, planning, and economic analysis -- detailed budget information, strategic plan, economic analysis of NIST programs
- Organizational chart -- shows NIST's organizational structure
- Staff directory -- use to get the phone number, mailing address, or email address of NIST employees

cookie disclaimer/privacy policy | descriptive text for the visually impaired

- A-Z subject index -- alphabetical listing of research topics, activities, programs, products, and services
- No Fear Act policy
- Small business contact
- Visitor information
- NIST Director's Office

About the Department of Commerce

- Strategic plan and annual performance plan
- Commerce and the Small Business Paperwork Reduction Act

NIST General Information

www.nist.gov/public_affairs/general_information.cfm

NIST Time | NIST Home | About NIST | Contact Us | A-Z Site Index

Search

Public Affairs Office

Publications | Subject Areas | Products/Services | NIST Organization | News | Programs & Projects | User Facilities | Work with NIST

NIST Home > Public Affairs Office > NIST General Information

Select Language

Powered by Google Translate

SHARE

NIST General Information

From the smart electric power grid and electronic health records to atomic clocks, advanced nanomaterials, and computer chips, innumerable products and services rely in way on technology, measurement, and standards provided by the National Institute of Standards and Technology.

Founded in 1901, NIST is a non-regulatory federal agency within the U.S. Department of Commerce. NIST's mission is to promote U.S. innovation and industrial competitiv advancing measurement science, standards, and technology in ways that enhance economic security and improve our quality of life.

NIST carries out its mission through the following programs:

- the NIST Laboratories, conducting world-class research, often in close collaboration with industry, that advances the nation's technology infrastructure and helps U.S. companies continually improve products and services;
- the Hollings Manufacturing Extension Partnership, a nationwide network of local centers offering technical and business assistance to smaller manufacturers to help them create and retain jobs, increase profits, and save time and money; and
- the Baldrige Performance Excellence Program, which promotes performance excellence among U.S. manufacturers, service companies, educational institutions, healt care providers, and nonprofit organizations; conducts outreach programs; and manages the annual Malcolm Baldrige National Quality Award which recognizes performance excellence and quality achievement;
- From 2007 to 2011, NIST provided cost-shared grants through the Technology Innovation Program, and between 1990 and 2007, it managed the Advanced Technolo Program.

NIST's FY 2014 resources total $850.0 million in direct appropriations, an estimated $47.3 million in service fees, and $107.0 million from other agencies.

The agency operates in two locations: Gaithersburg, Md., (headquarters—234-hectare/578-acre campus) and Boulder, Colo., (84-hectare/208-acre campus). NIST employs 3,000 scientists, engineers, technicians, and support and administrative personnel. NIST also hosts about 2,700 associates from academia, industry, and other government agencies, who collaborate with NIST staff and access user facilities. In addition, NIST partners more than 1,300 manufacturing specialists and staff at more than 400 MEP s locations around the country.

Our Organization

- Advancing Innovation and Industrial Competitiveness
- Mission, Vision, Core Competencies, and Core Values
- Organizational Chart
- Fact Sheets
- Employment Information
- NIST Tours
- Office of Information Systems Management
- Office of Financial Resources Management
- Office of Workforce Management

NIST General Information

From the smart electric power grid and electronic health records to atomic clocks, advanced nanomaterials, and computer chips, innumerable products and services rely in some way on technology, measurement, and standards provided by the National Institute of Standards and Technology.

Founded in 1901, NIST is a non-regulatory federal agency within the U.S. Department of Commerce. NIST's mission is to promote U.S. innovation and industrial competitiveness by advancing measurement science, standards, and technology in ways that enhance economic security and improve our quality of life.

NIST carries out its mission through the following programs:

- the NIST Laboratories, conducting world-class research, often in close collaboration with industry, that advances the nation's technology infrastructure and helps U.S. companies continually improve products and services;
- the Hollings Manufacturing Extension Partnership, a nationwide network of local centers offering technical and business assistance to smaller manufacturers to help them create and retain jobs, increase profits, and save time and money; and
- the Baldrige Performance Excellence Program, which promotes performance excellence among U.S. manufacturers, service companies, educational institutions, health care providers, and nonprofit organizations; conducts outreach programs; and manages the annual Malcolm Baldrige National Quality Award which recognizes performance excellence and quality achievement;
- From 2007 to 2011, NIST provided cost-shared grants through the Technology Innovation Program, and between 1990 and 2007, it managed the Advanced Technology Program.

NIST's FY 2014 resources total $850.0 million in direct appropriations, an estimated $47.3 million in service fees, and $107.0 million from other agencies.

The agency operates in two locations: Gaithersburg, Md., (headquarters—234-hectare/578-acre campus) and Boulder, Colo., (84-hectare/208-acre campus). NIST employs about 3,000 scientists, engineers, technicians, and support and administrative personnel. NIST also hosts about 2,700 associates from academia, industry, and other government agencies, who collaborate with NIST staff and access user facilities. In addition, NIST partners more than 1,300 manufacturing specialists and staff at more than 400 MEP service locations around the country.

Our Organization

- Advancing Innovation and Industrial Competitiveness
- Mission, Vision, Core Competencies, and Core Values
- Organizational Chart
- Fact Sheets
- Employment Information
- NIST Tours
- Office of Information Systems Management
- Office of Financial Resources Management
- Office of Workforce Management

What We Do

- NIST: Did You Know ...
- For Kids
- NIST Educational Activities
- NIST Centennial
- NIST Image Gallery
- The Story of NIST -- online timeline of NIST's history from 1901 to the present

Quotes, Speeches, Testimony

- What Others Say About NIST
- Testimony by NIST/DOC officials to Congress
- Speeches by NIST Officials

Impacts, FAQs, Links

- NIST: Did You Know ...
- NIST Environmental Policy
- Frequently Asked Questions
- Links to Related Web Sites
- Freedom of Information Act procedures

*this number was updated on 4/25/13

NIST: Did you know...

- NIST's weights and measures services, a job assigned to the federal government in the Constitution, provide the basis for the fairness and efficiency of sales. These services underpin the efficiency of about half the U.S. economy, or about $7 trillion of the U.S. gross domestic product.
- Eighty percent of global merchandise trade is influenced by testing and other measurement-related requirements of regulations and standards—and that U.S. companies increasingly depend on NIST to help ensure access to global markets that create new business and jobs?
- Without NIST, U.S. manufacturers of such products as glucose and cholesterol test kits—in vitro diagnostic (IVD) devices—wouldn't be able to meet requirements of new European Union regulations and would have been shut out of the $7 billion European market where they now have more than 60 percent of the business?
- The Malcolm Baldrige National Quality Award, the nation's highest honor awarded by the President of the United States to U.S. organizations for their performance excellence and quality achievements, is managed by NIST—and the award criteria are used by thousands of companies, schools, and hospitals to improve their products and services?
- The total economic benefit of the NIST Baldrige National Quality Program—which receives little federal funding—is estimated at almost $25 billion, for a stunning benefit-cost ratio of 207 to 1?
- The Baldrige program is so successful that it has spawned similar programs in more than 40 states?
- NIST-developed performance-measurement methods have accelerated the adoption of new manufacturing capabilities, such as accuracy-enhancing software for machine tools, and led to new technology, such as laser tracker measurement systems used increasingly in the aerospace, automotive, and other industries?
- NIST's technical contributions to the development of encryption standards for information technology have been estimated to have saved private industry more than billion—and enable consumers and business to be confident about the security of their
billions of dollars worth of electronic data transactions daily, such as withdrawals from ATMs?
- About 2.6 billion times a day, or 30,000 times a second, NIST's Internet Time Service sets computer clocks and other networked timekeeping devices, such as those used to synchronize telecommunications systems?
- NIST's Internet Time Service is used by NASDAQ members to time-stamp hundreds of billions of dollars worth of stock trades and other financial transactions conducted every business day?

NIST: Did you know...

- NIST's weights and measures services, a job assigned to the federal government in the Constitution, provide the basis for the fairness and efficiency of sales. These services underpin the efficiency of about half the U.S. economy, or about $7 trillion of the U.S. gross domestic product.
- Eighty percent of global merchandise trade is influenced by testing and other measurement-related requirements of regulations and standards—and that U.S. companies increasingly depend on NIST to help ensure access to global markets that create new business and jobs?
- Without NIST, U.S. manufacturers of such products as glucose and cholesterol test kits—in vitro diagnostic (IVD) devices—wouldn't be able to meet requirements of new European Union regulations and would have been shut out of the $7 billion European market where they now have more than 60 percent of the business?
- The Malcolm Baldrige National Quality Award, the nation's highest honor awarded by the President of the United States to U.S. organizations for their performance excellence and quality achievements, is managed by NIST—and the award criteria are used by thousands of companies, schools, and hospitals to improve their products and services?
- The total economic benefit of the NIST Baldrige National Quality Program—which receives little federal funding—is estimated at almost $25 billion, for a stunning benefit-cost ratio of 207 to 1?
- The Baldrige program is so successful that it has spawned similar programs in more than 40 states?
- NIST-developed performance-measurement methods have accelerated the adoption of new manufacturing capabilities, such as accuracy-enhancing software for machine tools, and led to new technology, such as laser tracker measurement systems used increasingly in the aerospace, automotive, and other industries?
- NIST's technical contributions to the development of encryption standards for information technology have been estimated to have saved private industry more than $1 billion—and enable consumers and business to be confident about the security of their
billions of dollars worth of electronic data transactions daily, such as withdrawals from ATMs?
- About 2.6 billion times a day, or 30,000 times a second, NIST's Internet Time Service sets computer clocks and other networked timekeeping devices, such as those used to synchronize telecommunications systems?
- NIST's Internet Time Service is used by NASDAQ members to time-stamp hundreds of billions of dollars worth of stock trades and other financial transactions conducted every business day?
- In the Army alone, 58,000 different types of equipment require regular "NIST traceable" calibration, ensuring that measurements made during the development, testing, maintenance, and operation are valid; that their system will perform as designed; and that they will be interoperable with other systems on the battlefield?
- NIST developed the testing procedures and other key technical elements of the Department of Homeland Security's performance standards for radiation detectors used at seaports, airports, and other points of entry into the United States?
- More than 3,000 law-enforcement officers have been spared from death or disabling injury as a result of NIST-developed standards for ballistic-resistant body armor ("bullet-proof" vests)?
- The National Research Council credits NIST Standard Reference Materials for DNA analyses with improving the accuracy of forensic DNA tests while reducing testing costs and that the criminal justice community depends on NIST to assure that its testing is accurate?
- The United States spends more than $2 trillion on health care, and 10 percent to 15 percent of that is associated with making measurements—NIST's specialty?
- NIST measurement services underpin the safety and effectiveness of about 10 million medical procedures that use radioactive materials—from prostate- and breast-cancer treatment to diagnostic imaging?
- NIST X-ray standards and proficiency tests ensure proper radiation exposure levels in more than 10,000 facilities that perform more than 30 million mammograms yearly?
- Many of the tools and materials used in modern dentistry—from the panoramic X-ray to composite fillings to an array of adhesives—originated at NIST through a partnership with the American Dental Association that began in 1928?
- NIST Standard Reference Materials helped to significantly improve the accuracy of cholesterol tests, resulting in better diagnoses and treatments and an estimated

annual savings of about $100 million?

- Consumers and industry benefit from NIST measurement aids that help manufacturers to validate food production and quality control procedures and to ensure accurate nutritional labeling?
- Emissions trading, called for by the Clean Air Act of 1990, depends on an initiative begun by NIST and the Environmental Protection Agency in cooperation with the specialty gas industry—and that a $64 billion market was created?
- NIST led the development of performance standards and placement recommendations for smoke detectors, now in 94 percent of American homes and saving thousands of lives, or that half of home fire deaths occur in the 6 percent of homes with no smoke alarms?
- NIST scientists have earned three Nobel Prizes over the last 11 years?
- Closed-captioning for people with impaired hearing, now featured on all TV sets, was co-invented at NIST, earning it an Emmy Award in 1980?
- Thousands of small manufacturers in all 50 states and Puerto Rico rely on the NIST Hollings Manufacturing Extension Partnership for hands-on technical and business assistance?
- In a recent survey completed in FY 2007, covering projects completed in fiscal year 2006, 4,959 clients of NIST's Hollings Manufacturing Extension Partnership reported that, as a result of MEP services, they realized $1.1 billion in cost savings; invested $1.6 billion in modernization, including plant and equipment, information systems, and workforce training; increased or retained $6.8 billion in sales; and created and retained over 52,000 jobs?
- U.S. automakers and their suppliers, which account for more than 3 percent of the nation's GDP, rely on NIST Standard Reference Materials to ensure quality in a wide range of tasks, from the processing of materials to final assembly, to emissions compliance?

The National Institute of Standards and Technology (NIST) is an agency of the U.S. Department of Commerce.

Privacy Policy / Security Notice / Accessibility Statement / Disclaimer / Freedom of Information Act (FOIA) / Environmental Policy Statement / No Fear Act Policy / NIST Information Quality Standards / Scientific Integrity Summary

Sign Up for NIST E-mail alerts:
Enter email address Go

Date created: October 2, 2008 | Last updated: October 5, 2010 Contact: **Webmaster**

Exhibit E

Open standard - Wikipedia,
en.wikipedia.org/wiki/Open_standard#Programming_languages
Create account Log in
Article Talk Read Edit View history Search
WIKIPEDIA
The Free Encyclopedia
Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikimedia Shop
Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact page
Tools
What links here
Related changes
Upload file
Special pages
Permanent link
Page information
Wikidata item
Cite this page
Print/export
Create a book
Download as PDF
Printable version
Languages
العربية
Català
Cymraeg
Dansk
Deutsch
Ελληνικά
Español
فارسی
Français
한국어
Bahasa Indonesia
Italiano

Open standard

From Wikipedia, the free encyclopedia

This article's **use of external links may not follow Wikipedia's policies or guidelines**. Please improve this article by removing excessive or inappropriate external links, and converting useful links where appropriate into footnote references. *(August 2012)*

An **open standard** is a standard that is publicly available and has various rights to use associated with it, and may also have various properties of how it was designed (e.g. open process). There is no single definition and interpretations vary with usage.

The terms "open" and "standard" have a wide range of meanings associated with their usage. There are a number of definitions of open standards which emphasize different aspects of openness, including the openness of the resulting specification, the openness of the drafting process, and the ownership of rights in the standard. The term "standard" is sometimes restricted to technologies approved by formalized committees that are open to participation by all interested parties and operate on a consensus basis.

The definitions of the term "open standard" used by academics, the European Union and some of its member governments or parliaments such as Denmark, France, and Spain preclude open standards requiring fees for use, as do the New Zealand, South African and the Venezuelan governments. On the standard organisation side, the World Wide Web Consortium (W3C) ensures that its specifications can be implemented on a royalty-free basis.

Many definitions of the term "standard" permit patent holders to impose "reasonable and non-discriminatory licensing" royalty fees and other licensing terms on implementers and/or users of the standard. For example, the rules for standards published by the major internationally recognized standards bodies such as the IETF, ISO, IEC, and ITU-T permit their standards to contain specifications whose implementation will require payment of patent licensing fees. Among these organizations, only the IETF and ITU-T explicitly refer to their standards as "open standards," while the others refer only to producing "standards." The IETF and ITU-T use definitions of "open standard" that allow "reasonable and non-discriminatory" patent licensing fee requirements.

There are those in the open-source software community who hold that an "open standard" is only open if it can be freely adopted, implemented and extended.[1] While open standards or architectures are considered non-proprietary in the sense that the standard is either un-owned or owned by a collective body it can still be publicly shared and not tightly guarded.[2] The typical example of "open source" that has become a standard is the personal computer originated by IBM and now referred to as Wintel, the combination of the Microsoft operating system and Intel microprocessor. There are three others that are most widely accepted as "open" which include the GSM phones (adopted as a government standard), Open Group which promotes UNIX and the like, and the Internet Engineering Task Force (IETF) which created the first standards of SMTP and TCP/IP. Buyers tend to prefer open standards which they believe offer them cheaper products and more choice for access due to network effects and increased competition between vendors.[3]

Open standards which specify formats are sometimes referred to as open formats.

Many specifications that are sometimes referred to as standards are proprietary and only available under restrictive contract terms (if they can be obtained at all) from the organization that owns the copyright on the specification. As such these specifications are not considered to be fully *Open*. Joel West has argued that "open" standards are not black and white but have many different levels of "openness". Ultimately a standard needs to be open enough that it will become adopted and accepted in the market, but still closed enough that firms can get a return on their investment in developing the technology around the standard. A more open standard tends to occur when the knowledge of the

Exhibit F

OpenCL for OS X

OpenCL lets you tap into the parallel computing power of modern GPUs and multicore CPUs to accelerate compute-intensive tasks in your Mac apps. Use OpenCL to incorporate advanced numerical and data analytics features, perform cutting-edge image and media processing, and deliver accurate physics and AI simulation in games.

Reference, Guides, and Sample Code

Selecting a GPU for OpenCL on the Mac Pro (Late 2013)
Learn how to query the GPUs of the latest Mac Pro and select one or both for use by OpenCL.

OpenCL and OpenGL Support in OS X Mavericks
Get details on the OpenCL and OpenGL versions supported by Mac computers running OS X Mavericks.

OS X v10.9 API Diffs
Review the latest updates to the OpenCL.framework for OS X Mavericks.

OpenCL Programming Guide for Mac
Learn about OpenCL for OS X, including parallelizable routines, Grand Central Dispatch, memory objects, data sharing, and more.

OpenCL Specification
Get full details on the OpenCL standard from the OpenCL Specification version 1.2.

OpenCL N-Body Simulation Example
Uses OpenCL and OpenGL to simulate a classical N-Body gravitational system while efficiently using one or more GPUs.

OpenCL Hello World Example
View sample code on basic usage of OpenCL, calculating a mathematical square for a buffer of floating point values.

OpenCL Procedural Geometric Displacement Example
View sample code on binding OpenCL to existing OpenGL buffers.

OpenCL Procedural Grass and Terrain Example
View sample code on creating a procedural field of grass on a generated terrain model, which is then rendered with OpenGL.

Videos

Learn about OpenCL with video presentations and tutorials.

- Harnessing the Power of the Mac Pro with OpenGL and OpenCL
- Working with OpenCL
- What's New in OpenGL for OS X
- Core Image Effects and Techniques

Forums

Post questions and discuss OpenCL with other developers and Apple engineers.
View Forums

Using the OpenCL Logo

If your app uses OpenCL or you've created your own implementation, learn about licensing the OpenCL logo.

Tools and Related Resources

- OpenGL for OS X
- Xcode

OpenCL Hello World Exampl

Apple Inc. [US] https://developer.apple.com/library/mac/samplecode/OpenCL_Hell

Mac Developer Library

Developer

OpenCL Hello World Example

Table of Contents

Download Sample Code

About OpenCL Hello World Example

ReadMe.txt

hello.c

Revision History

Next

OpenCL Hello World Example

Last Revision:	Version 1.4, 2011-03-01 OpenCL kernel was missing void return qualifier (Full Revision History)
Build Requirements:	Mac OS X 10.6, Xcode 3.2
Runtime Requirements:	Mac OS X 10.6

A simple "Hello World" compute example showing basic usage of OpenCL which calculates the mathematical square (X[i] = pow(X[i],2)) for a buffer of floating point values.

Next

Copyright © 2011 Apple Inc. All Rights Reserved. Terms of Use | Privacy Policy | Updated: 2011-03-01

Feedback

Table of Contents

- Introduction
- Hello World!
- Basic Programming Sample
- OpenCL On OS X Basics
- Identifying Parallelizable Routines
- How the Kernel Interacts With Data in OS X OpenCL
- Using Grand Central Dispatch With OpenCL
- Memory Objects in OS X OpenCL
- Creating and Managing Image Objects In OpenCL
- Creating and Managing Buffer Objects In OpenCL
- Sharing Data Between OpenCL and OpenGL
- Controlling OpenCL / OpenGL Interoperation With GCD
- Using IOSurfaces With OpenCL
- Autovectorizer
- Tuning Performance On the GPU
- Improving Performance On the CPU
- Binary Compatibility Of OpenCL Kernels
- Revision History

Next

About OpenCL for OS X

OpenCL™ (Open Computing Language) is an open standard for cross-platform, programming of modern highly-parallel processor architectures. Introduced with OS X v10.6, OpenCL consists of a C99-based programming language designed for parallelism, a powerful scheduling API, and a flexible runtime that executes kernels on the CPU or GPU. OpenCL lets your application harness the computing power of these processors to improve performance and deliver new features based on compute-intensive algorithms.

In addition to support for the OpenCL 1.1 standard, OS X v10.7 adds integration between OpenCL, Grand Central Dispatch (*GCD*), and Xcode to make it even easier to use OpenCL in your application.

At a Glance

Using OpenCL is easier than ever as of OS X v10.7:

- OpenCL is fully supported by Xcode. The Xcode offline compiler removes a configuration step that used to have to be performed before the kernel could be run and facilitates debugging earlier in the development process. See Hello World!.
- You can write OpenCL functions in separate files and include them in your Xcode project. You can compile the kernels when your application is built, before it runs. This improves runtime performance.
- OpenCL now integrates with GCD, making it easier for you to focus on making your OpenCL kernels more efficient. See Using Grand Central Dispatch With OpenCL.
- The autovectorizer compiles and accelerates performance of kernels that run on the CPU up to four times without additional effort. The autovectorizer allows you to write one kernel that runs efficiently on both a CPU and a GPU. You can invoke the autovectorizer regardless of whether you are compiling from Xcode or building the kernels at runtime. Or you can disable the autovectorizer if necessary. See Autovectorizer.

You can, of course, continue to use code you've already written to the OpenCL 1.1 standard. But see Binary Compatibility Of OpenCL Kernels for a note about how to handle existing binaries.

Feedback

Exhibit G

Exhibit H

United States of America

United States Patent and Trademark Office

MAMDA

Reg. No. 4,288,465

Registered Feb. 12, 2013

Int. Cl.: 9

TRADEMARK

PRINCIPAL REGISTER

NYSE TECHNOLOGIES, INC. (DELAWARE CORPORATION)
11 WALL STREET
NEW YORK, NY 10005

FOR: COMPUTER SOFTWARE, NAMELY, APPLICATION PROGRAMMING INTERFACE (API), FRAMEWORK AND SUPPORTING LIBRARIES THAT PROVIDE SUPPORT FOR INTERPRETING MARKET DATA MESSAGES AND ALLOW APPLICATIONS TO CONSUME THE DATA THROUGH DOMAIN SPECIFIC INTERFACES AND CLASSES FOR TRADING CONSTRUCTS, INCLUDING TRADES, QUOTES, SECURITY STATUS, FUNDAMENTAL DATA, ORDER IMBALANCE INFORMATION, ORDER BOOKS, AND OPTIONS CHAINS, IN CLASS 9 (U.S. CLS. 21, 23, 26, 36 AND 38).

FIRST USE 3-28-2007; IN COMMERCE 3-28-2007.

THE MARK CONSISTS OF STANDARD CHARACTERS WITHOUT CLAIM TO ANY PARTICULAR FONT, STYLE, SIZE, OR COLOR.

SER. NO. 85-643,319, FILED 6-5-2012.

MATTHEW MCDOWELL, EXAMINING ATTORNEY

Acting Director of the United States Patent and Trademark Office

Trademarks > Trademark Electronic Search System (TESS)

TESS was last updated on Tue Mar 10 03:20:59 EDT 2015

Record 1 out of 1

MAMDA

Word Mark	MAMDA
Goods and Services	IC 009. US 021 023 026 036 038. G & S: Computer software, namely, application programming interface (API), framework and supporting libraries that provide support for interpreting market data messages and allow applications to consume the data through domain specific interfaces and classes for trading constructs, including trades, quotes, security status, fundamental data, order imbalance information, order books, and options chains. FIRST USE: 20070328. FIRST USE IN COMMERCE: 20070328
Standard Characters Claimed	
Mark Drawing Code	(4) STANDARD CHARACTER MARK
Serial Number	85643319
Filing Date	June 5, 2012
Current Basis	1A
Original Filing Basis	1A
Published for Opposition	November 27, 2012
Registration Number	**4288465**
Registration Date	February 12, 2013
Owner	(REGISTRANT) NYSE Technologies, Inc. CORPORATION DELAWARE 11 Wall Street New York NEW YORK 10005
Attorney of Record	Marylee Jenkins
Type of Mark	TRADEMARK
Register	PRINCIPAL
Live/Dead Indicator	LIVE

Int. Cl.: 9

Prior U.S. Cls.: 21, 23, 26, 36, and 38

Reg. No. 3,677,861

United States Patent and Trademark Office

Registered Sep. 1, 2009

TRADEMARK
PRINCIPAL REGISTER

PXML

PRINTRONIX, INC. (DELAWARE CORPORATION)
14600 MYFORD ROAD
IRVINE, CA 926061005

FOR: SOFTWARE, NAMELY, AN APPLICATION PROGRAMMING INTERFACE FOR PRINTERS TO ENABLE SOFTWARE PROGRAMMERS TO INTEGRATE PRINTER MANAGEMENT FEATURES INTO COMPUTER SOFTWARE , IN CLASS 9 (U.S. CLS. 21, 23, 26, 36 AND 38).

FIRST USE 9-0-2005; IN COMMERCE 9-0-2005.

THE MARK CONSISTS OF STANDARD CHARACTERS WITHOUT CLAIM TO ANY PARTICULAR FONT, STYLE, SIZE, OR COLOR.

SN 78-645,569, FILED 6-7-2005.

BERYL GARDNER, EXAMINING ATTORNEY

Trademarks > Trademark Electronic Search System (TESS)

TESS was last updated on Tue Mar 10 03:20:59 EDT 2015

Record 1 out of 1

PXML

Word Mark	**PXML**
Goods and Services	IC 009. US 021 023 026 036 038. G & S: Software, namely, an application programming interface for printers to enable software programmers to integrate printer management features into computer software. FIRST USE: 20050900. FIRST USE IN COMMERCE: 20050900
Standard Characters Claimed	
Mark Drawing Code	(4) STANDARD CHARACTER MARK
Trademark Search Facility Classification Code	LETTER-3-OR-MORE PXML Combination of three or more letters as part of the mark
Serial Number	78645569
Filing Date	June 7, 2005
Current Basis	1A
Original Filing Basis	1B
Published for Opposition	June 24, 2008
Registration Number	3677861
Registration Date	September 1, 2009
Owner	(REGISTRANT) Printronix, Inc. CORPORATION DELAWARE 15345 Barranca Parkway Irvine CALIFORNIA 92618
Assignment Recorded	ASSIGNMENT RECORDED
Attorney of Record	David A. Bell
Type of Mark	TRADEMARK
Register	PRINCIPAL
Live/Dead Indicator	LIVE

United States of America

United States Patent and Trademark Office

DynamicOSI

Reg. No. 4,280,967

Registered Jan. 22, 2013

Int. Cl.: 9

TRADEMARK

SUPPLEMENTAL REGISTER

MONFOX LLC (GEORGIA LIMITED LIABILITY COMPANY), DBA MONFOX,
3840 WINDERMERE PARKWAY
SUITE 403
CUMMING, GA 30041

FOR: PLATFORM INDEPENDENT APPLICATION PROGRAMMING INTERFACE (API) SOFTWARE FOR MANAGEMENT OF TRANSPORT, PRESENTATION AND SESSION LAYERS AS DEFINED IN AN OPEN SYSTEMS INTERCONNECTION (OSI) MODEL, IN CLASS 9 (U.S. CLS. 21, 23, 26, 36 AND 38).

FIRST USE 1-31-2011; IN COMMERCE 1-31-2011.

THE MARK CONSISTS OF STANDARD CHARACTERS WITHOUT CLAIM TO ANY PARTICULAR FONT, STYLE, SIZE, OR COLOR.

SER. NO. 85-569,641, FILED P.R. 3-14-2012; AM. S.R. 12-3-2012.

ASMAT KHAN, EXAMINING ATTORNEY

David J. Kappos

Director of the United States Patent and Trademark Office

Trademarks > Trademark Electronic Search System (TESS)

TESS was last updated on Tue Mar 10 03:20:59 EDT 2015

Record 1 out of 1

TSDR | ASSIGN Status | TTAB Status ***(Use the "Back" button of the Internet Browser to return to TESS)***

DynamicOSI

Word Mark	**DYNAMICOSI**
Goods and Services	IC 009. US 021 023 026 036 038. G & S: Platform independent application programming interface (API) software for management of transport, presentation and session layers as defined in an open systems interconnection (OSI) model. FIRST USE: 20110131. FIRST USE IN COMMERCE: 20110131
Standard Characters Claimed	
Mark Drawing Code	(4) STANDARD CHARACTER MARK
Serial Number	85569641
Filing Date	March 14, 2012
Current Basis	1A
Original Filing Basis	1A
Date Amended to Current Register	December 3, 2012
Registration Number	4280967
Registration Date	January 22, 2013
Owner	(REGISTRANT) Monfox LLC DBA Monfox LIMITED LIABILITY COMPANY GEORGIA 3840 Windermere Parkway Suite 403 Cumming GEORGIA 30041
Attorney of Record	Christopher J. Day
Type of Mark	TRADEMARK
Register	SUPPLEMENTAL
Live/Dead Indicator	LIVE

United States of America

United States Patent and Trademark Office

MxLWare

Reg. No. 3,854,657

Registered Sep. 28, 2010

Int. Cl.: 9

TRADEMARK

PRINCIPAL REGISTER

MAXLINEAR, INC. (CALIFORNIA CORPORATION)
2051 PALOMAR AIRPORT ROAD, SUITE 100
CARLSBAD, CA 92011

FOR: COMPUTER APPLICATION SOFTWARE FOR MOBILE PHONES; COMPUTER FIRMWARE FOR PROVIDING A COMMON APPLICATION PROGRAMMING INTERFACE AMONG MULTIPLE SOFTWARE SERVICES AND ABSTRACTING MULTIPLE HARDWARE DEVICES FUNCTIONALITY TO A COMMON FRAMEWORK; COMPUTER SOFTWARE FOR WIRELESS CONTENT DELIVERY, IN CLASS 9 (U.S. CLS. 21, 23, 26, 36 AND 38).

FIRST USE 10-9-2009; IN COMMERCE 10-9-2009.

THE MARK CONSISTS OF STANDARD CHARACTERS WITHOUT CLAIM TO ANY PARTICULAR FONT, STYLE, SIZE, OR COLOR.

NO CLAIM IS MADE TO THE EXCLUSIVE RIGHT TO USE "WARE", APART FROM THE MARK AS SHOWN.

SN 77-762,558, FILED 6-17-2009.

CHRISTINA SOBRAL, EXAMINING ATTORNEY

David J. Kappos

Director of the United States Patent and Trademark Office

Trademarks > Trademark Electronic Search System (TESS)

TESS was last updated on Tue Mar 10 03:20:59 EDT 2015

Record 1 out of 1

MxLWare

Word Mark	MXLWARE
Goods and Services	IC 009. US 021 023 026 036 038. G & S: Computer application software for mobile phones; Computer firmware for providing a common application programming interface among multiple software services and abstracting multiple hardware devices functionality to a common framework; Computer software for wireless content delivery. FIRST USE: 20091009. FIRST USE IN COMMERCE: 20091009
Standard Characters Claimed	
Mark Drawing Code	(4) STANDARD CHARACTER MARK
Trademark Search Facility Classification Code	LETTER-3-OR-MORE MXL-WARE Combination of three or more letters as part of the mark
Serial Number	77762558
Filing Date	June 17, 2009
Current Basis	1A
Original Filing Basis	1B
Published for Opposition	November 24, 2009
Registration Number	**3854657**
Registration Date	September 28, 2010
Owner	(REGISTRANT) MaxLinear, Inc. CORPORATION CALIFORNIA 2051 Palomar Airport Road, Suite 100 Carlsbad CALIFORNIA 92011
Disclaimer	NO CLAIM IS MADE TO THE EXCLUSIVE RIGHT TO USE "WARE" APART FROM THE MARK AS SHOWN
Type of Mark	TRADEMARK
Register	PRINCIPAL
Live/Dead Indicator	LIVE

Exhibit I

MAMDA API Developer's Guide

API Developer's Guide for MAMDA API

October 29, 2010

Copyright 2010 NYSE Technologies Inc. All rights reserved. All third party trademarks are owned by their respective owners and are used with permission. NYSE Technologies Inc. and its affiliates do not recommend or make any representation as to possible benefits from any securities, or investments or third party products or services. Investors should undertake their own due diligence regarding their securities and investment practices. This document may contain forward-looking statements regarding NYSE Technologies Inc. and their affiliates. Such statements are based upon the current beliefs and expectations of management and are subject to significant risks and uncertainties. Actual results may differ from those set forth in the forward-looking statements. This document contains a link to a third-party website. NYSE Technologies Inc. and its affiliates are not responsible for the content of such website, nor have they developed, checked for accuracy or otherwise reviewed the content or privacy policy of any such web site.

Table of Contents

1. Introduction 3
  1.1. Document Conventions 4
  1.2. General 4
  1.3. The core classes within a MAMDA application 4
  1.4. The libraries 6
  1.5. The examples 6
2. Writing a simple MAMDA application 8
  2.1. The role of MAMA in a MAMDA application 8
  2.2. The building blocks of a MAMDA application 8
3. Simple market data classes 10
  3.1. Quotes 10
    3.1.1. Event notifications 13
    3.1.2. Accessing Quote data 13
  3.2. Trades 14
    3.2.1. Event notifications 17
    3.2.2. Accessing trade data 18
  3.3. Order Imbalance 19
  3.4. Security Status 19
  3.5. Fundamental data 19

4. Structured market data classes 19
4.1. Order Books 19
4.1.1. The Wombat Order Book 19
4.1.2. WombatMsg and order books 21
4.1.3. Why no order book support in MAMA? 21
4.1.4. Order books in MAMDA 22
4.2. Order Book Publishing 39
4.2.1. Creating an order book publishing example 40
4.3. Options Chains 42
4.3.1. Accessing the chain and contract information 45
4.3.2. Sliding window option chain views 47
4.4. News 47
4.4.1. News Data 48
5. Non market data utility classes 50
5.1. MamdaMultiParticipantManager 51
5.2. MamdaMultiSecurityManager 53
Appendix A. Changes between 3.2 and 4.0 55
A.1. Interface Changes 55
A.1.1. MamdaSession 55
A.1.2. MamdaStaleListener 56
A.1.3. Size/Volume Fields 56
A.1.4. OrderBooks 56
A.2. New/Altered Functionality 59
A.2.1. News 60
A.2.2. OrderBooks 60
A.2.2.1. Behavior changes 60
A.2.2.2. New functionality 60

1. Introduction

Traditional application programming interfaces (APIs) & used for market data distribution provides anonymous field and/or record based interfaces. These APIs require applications to iterate over fields and provide internal logic to determine what type of message the collection of fields represents. Wombat's Middleware Agnostic Messaging API (MAMA) is one of the traditional APIs. For some applications like GUIs, the application need not understand the context of the collection of fields; they only need to display each field on a particular part of the screen. Other applications, like program trading, tick capture, analytical, and smart order routing applications need a deeper understanding of the meaning and context of each field in the message as well as the type of message the update represents.

Classifying market data messages extends beyond determining the messages' high level types like quotes, trades, order book updates, security status updates, fundamental data, etc. Applications must further categorize these messages into subtypes: regional vs consolidated quotes, regional vs consolidated trades, trade cancels/errors/corrections, out of order trades etc. The field based approach to accessing market data demands a great deal of complex logic to correctly classify incoming data. Wombat's Middleware Agnostic Market Data API (MAMDA) augments MAMA with a rich set of market data related data structures that address the shortcomings of field based APIs. The object oriented MAMDA, C++ and Java, APIs perform the tedious work of deciphering field based data from MAMA into various types of messages and provides convenient interfaces for applications to process this data.

MAMDA provides for the following functionality:

- Callback based API that separates incoming data into appropriate classes and subclasses of event.
  - Trades
  - Quotes
  - Security Status
  - Fundamental data
  - Order imbalance information
  - Order books
  - Options chains
  - News
- Maintains an internal cache and updates it as data arrives.
- Flexible listener based API for simple application development.

This document describes MAMDA 3.3.x version of the API but the many of the principles discussed apply equally to older versions of the API.

1.1. Document Conventions

The term 'object' is used throughout to refer to C structures and C++/Java object types interchangeably.

The term 'symbol' is used throughout the document and is interchangeable with the concept of a 'topic' which may be more common to developers familiar to publish/subscribe messaging.

1.2. General

This C++ and Java implementations of the MAMDA API expose the same top-level objects and interfaces with minor language dependant interface differences. This document describes the C++ API in detail and notes major differences in the Java API when applicable. All examples are illustrated using C++. Specific Java examples are provided where clarity on any differences between the two implementations is required. In general, however, both flavours of the API provide an identical interface.

Several MAMDA objects require a two step initialization process where the caller first allocates the object and then initializes it by invoking a create() method. This is required where memory allocation and creation are two logically different steps and where properties on an object can be set/changed prior to creation.

```
MamdaSubscription subsc  = new MamdaSubscription();

... /*Set properties on the subscription*/

subsc->create(...);
```

MAMDA applications all require three core classes: a subscription, one or more listeners, and one or more handlers. Applications use the pattern to process quotes, trades, order books, news, etc. Subsequent sections of this document address this pattern and its application in detail.

Both the C++ and Java MAMDA APIs propagate errors through a combination of exceptions and callbacks. MAMDA uses callbacks in cases, like message loops, where catching exceptions is not practical. The C++ API throws MAMDA specific exception that derive from the **C++ Standard Library** exception classes like `invalid_argument`, etc. The Java API throws exceptions derived from `java.lang.RuntimeException` and `com.wombat.mama.MamaException`.

All functions/methods in the API, across all language implementations and transports, exhibit the same behaviour unless otherwise stated.

Source code and sample build files for example programs are available in the examples/mamda directory of the Wombat API distribution.

1.3. The core classes within a MAMDA application

The MamdaSubscription, Mamda<type>Listener, Mamda<type>Handler and Mamda<type>Fields are the four core divisions of classes used in the writing of a MAMDA application.

- MamdaSubscription:

Used to register interest in an instrument. The MamdaSubscription replaces the need for the MamaSubscription when subscribing to market data on the Wombat Market Data Platform. To process data returned via a MamdaSubscription instance one or more MamdaMsgListeners (and usually handlers) must be registered (see below).

- Mamda<type>Listener

  The Mamda<data type>Listener classes are responsible for identifying the type of data represented by an incoming MamaMsg. Each of the Listener classes, with the exception of the MamdaMsgListener, then identifies the event type for the message type and invokes appropriate event callbacks on any registered Mamda<type>Handler class instances (see below).

  The Listener objects also maintain an up to date cache of all updates received so far. This provides the application with ready access to the latest state for each instrument without having to contact the feeds. As a result, listener instances must not be reused across multiple subscriptions.

  Each of the Mamda<type>Listeners derives from the MamdaMsgListener super class (which can be used in its own right for the most basic of applications).

  Examples of MamdaMsgListener sub classes are:

  - MamdaQuoteListener
  - MamdaTradeListener
  - MamdaOrderImbalanceListener
  - MamdaSecStatusListener
  - MamdaFundamentalListener
  - MamdaOrderBookListener

- Mamda<type>Handler

  The Mamda<type>Handler callback classes are used to propagate event notifications related to the type of messages processed by the parent Mamda<type>Listener class. Concrete implementations of these interfaces exist for each of the market data types supported by the API.

- Mamda<type>Fields

  To identify data from messages MAMDA uses multiple caches of data dictionary field descriptors, one per market data type and a single common set of fields. It is the responsibility of an application built using the API to obtain the data dictionary from the platform and to initialize each of the field caches required depending on the market data type being subscribed to. Each of these caches need only be initialized once for an application. Failure to initialize the appropriate cache for a market data listener can result in unexpected behaviour. In addition to the market data object specific class there exists a MamdaCommonFields class which contains field descriptors shared across objects within the API, this cache should be initialized for all applications regardless of the market data being looked at.

Note: The Mamda<Type>Fields classes simply maintain pointers/references to the field descriptors in the data dictionary. As such, it is necessary to keep a reference to the data dictionary for the lifetime of the MAMDA based application.

Versions 4.0.18r and onwards also have a reset method for the Mamda<Type>Fields classes. Fields then need to be initialized once more. This allows the fields to be reset during the lifetime of the Mamda application without restarting the application.

1.4. The libraries

Each of the key functional areas of the MAMDA API comes bundled in a separate library. These are listed and described in the table below:

Library	Description
libmamda.[a][so]/libmamda.jar	The core foundation classes of the API including all record based listeners and event/data objects.
libmamdabook.[a][so]/mamda_book.jar	All classes required for the MAMDA orderbook processing. Includes atomic books for the C++ API. Depends on libmamda.
libmamdaoptions.[a][so]/mamda_options.jar	All classes required for the MAMDA options processing. Depends on libmamda.
libmamdanews.[a][so]	All classes required for the MAMDA news processing. Depends on libmamda. (C++ only)

1.5. The examples

The MAMDA API comes bundled with a number of example programs illustrating the use of various aspects of the API.

Example	Description
mamdalisten/MamdaListen	The most basic MAMDA application simply using the MamdaMsgListener. Simply prints the contents of each message received to the console window. E.g. mamdalisten -S NASDAQ -s MSFT -tport tport_name
quoteticker/MamdaQuoteTicker	Illustrates using the MamdaQuoteListener and related classes. Prints bid/ask size and price for each quote update received. E.g. quoteticker -S NASDAQ -s MSFT -tport tport_name
enterprisequoteticker.cpp (C++ only)	As above using managed subscriptions.

PRINTRONIX®

PXML Application Developer Reference Manual

PXML Application Developer Reference Manual

Trademark Acknowledgments

Printronix and PSA are registered trademarks of Printronix, Inc.

PXML, SL4M, T4M, SL5000r, and T5000r are trademarks of Printronix, Inc.

COPYRIGHT 2006, 2008 PRINTRONIX, INC.
All rights reserved.

Table of Contents

1 Overview 7

Introduction 7

Constraints 8

2 PXML 9

PXML Selection 9

PTX_SETUP Commands To Activate PNE (Version 1.0) 9

PTX_SETUP Commands To Activate UCP (Version 2.0) 10

PTX_SETUP Commands To Activate PXML 10

Transport 11

Messaging 11

Message Root 11

Password Security 12

Message Unique ID 12

Status 13

Job Markers 13

Get Status Request 14

Automatic Status Select 14

Job Status Reply Message 15

Label Response Message 16

RFID Tag Response Message 17

Display Response Message 21

Fault Response Message 21

Engine Status Response Message 22

Statistics 22

RFID Statistic Get Request 22

RFID Statistic Clear Request 23

RFID Statistic Response 23

Reboot Request 24

Information Request 25

Information Get Request 25

Printer Information Response 25

Server Information Response 27

rfidTagOption Information Response 27

Printer ACK or Error Response 28

PXML Inactive Error Response 29

Printer Setting 29
Printer Setting Get Request 29
Printer Setting Select Request For rfidSetting 29
Time Setting Response 30
Time Set Request 30
PGL Error Handling Response 30
PGL Error Handling Set Request 31
Network Setting Response 31
Network Set Request 33
RFID Setting Response 34
RFID Set Request 34
Printer Configuration 36
Load Configuration Request 36
Save Configuration Request 36
Save Configuration Response 37
Printer Storage 37
Flash or EMC Storage Request 37
Flash or EMC File System Directory Response 37
Flash or EMC Delete File Request 38
Flash or EMC Read File Request 39
FLASH or EMC Read File Response 39
Alerts 40
Validate Requests And Responses Using XML Schema 45

A New Features In PXML 47
Version 2.0 47
Version 2.1 47

B Formal PXML Schema 49

C Contact Information 71
Printronix Customer Support Center 71
Printronix Supplies Department 71
Corporate Offices 72

1 *Overview*

Introduction

This manual describes a communication model for remotely managing Printronix printers. It is for application developers to implement custom printer management schemes.

Two facets to the management model include:

- management messages and the operations and events that they describe
- the transport mechanism that moves the messages back and forth between the printer and the client software.

The main component of the model is the Printronix XML Device Management Language, which describes the management and response messages. PXML™ is an XML based language that allows a client application to issue commands to a printer and receive responses from the printer. Responses can be either a response to a command (solicited responses), or responses generated by events that occur during printer operation (unsolicited responses).

Please visit the Printronix website (www.printronix.com) to view the printer models and firmware versions available with PXML.

PXML provides the following features:

- printer power-up modes – reset and download.
- different types of printer status reporting.
- job status reporting, including RFID tag and online data validation information.
- configuration change.
- menu changes.

Constraints

- UCP (for GPIO) and PXML cannot be used at the same time in the system. Only one can be active at any time.
- Only one client may connect to the management port at a time.
- The client is responsible for maintaining the TCP management connection to the printer; reconnect as necessary. The management connection may be lost at any time due to printer power-down or network failure. Responses will be lost when the management connection is down since there is no caching or retransmission mechanism.
- It is possible for a client to connect to the management port in the middle of a response message, which at the same time is being sent to a previous client. This is possible if the connection to the previous client is dropped before the message is complete. It is the responsibility of the new client to read and discard the remainder of the message.
- All responses are not synchronized, so a message immediately returned after a command may not be the one that applies to the command. The messages are in XML format and are guaranteed to be complete messages, except when the management connection is lost and reestablished while a message was being sent from the printer.
- XML messages must start with the <?xml> header so that invalid messages can be ignored by the printer.
- The TCP transport mechanism does not provide a separate framing protocol, so the input stream must be processed by an XML parser to determine the end of a message. Alternatively, message framing can be determined by searching for the <?xml> header, then capturing all data until the </pxml> trailer is found.
- One response may span more than one message to relieve the printer from having to buffer output messages. Messages that form a single response are guaranteed to be sent in the correct sequence, and a flag is provided to denote the final message of the sequence.
- For firmware download, the client is responsible for waiting for the printer to enter download mode before sending data. The client is responsible for waiting until the printer fully boots with the new firmware before sending configuration data.
- Notification of start and end of print job must be initiated by some other means, typically be placing PTX_SETUP commands in the print data stream.

2 PXML

PXML Selection

PrintNet Enterprise (PNE) is the Printronix proprietary management protocol used by the Printronix PrintNet Enterprise software. PXML is the XML based management protocol that serves as a direct interface between third party software. In version 1.0, the printer factory default is PNE, but the user can select which management protocol to use via PTX_SETUP commands or from the front panel. PNE and PXML can't be use concurrently in the system, only one can be active at any time. In version 2.0 and above, both PNE and PXML are active at the same time. However, PXML and GPIO (UCP) share the same channel and only one can be active at any time in version 2.0 and above.

For the user connecting to the printer using third party application, there is an initial setup. The initial setup is done with a PTX_SETUP file sent through the printing port. This is only necessary for version 1.0 of PXML. The PTX_SETUP file involves the following steps:

- Select PXML as management protocol
- Select Ethernet to be the interface for PXML
- Set 3007 to be the PXML Port number

You can also use the front panel menu options to perform these steps.

To switch from PXML to PPM:

- Select PNE as management protocol
- Set 3001 to be the PNE Port number

These steps can be done with a PTX_SETUP file or from changing the front panel menu. A printer power cycle will be performed after the PNE or PXML port number change and goes back online. The change stays persistent in the printer.

PTX_SETUP Commands To Activate PNE (Version 1.0)

NOTE: Only necessary for version 1.0 of PXML support.

This sets up PNE using the Ethernet port on port number 3001 for version 1.0:

```
!PTX_SETUP
CONFIG-MPI_SELECT;PNE
CONFIG-PNE_PORT;ETHERNET
CONFIG-PNE_PORT_NUM;3001
PTX_END
```

```
-----------------------------------------------------------------
-- DynamicOSI(tm) Release 1_3_1: RELEASE NOTES
-----------------------------------------------------------------

-- ACSE API: add AE support for AE/AP responding
             invocation identifier


   - The responding AE/AP invocation identifier parameters in
     the AARE ACSE response were ignored from the AEParams since
     they were deemmed general unnecessary in the AARE response.

     Now, if you provide a non-negative integer value for either
     of these parameters in the RFC1006AEParams for the stack,
     the value will be included in AARE message.

-----------------------------------------------------------------
-- DynamicOSI(tm) Release 1_2_3: RELEASE NOTES
-----------------------------------------------------------------

-- Fixed problem with Javadoc

-----------------------------------------------------------------
-- DynamicOSI(tm) Release 1_2_2: RELEASE NOTES
-----------------------------------------------------------------

-- Logger API

   Enhancement: Added 'autoRecreate' feature to the logger which
                will cause the logger to automatically
                recreate the log file if it is deleted at runtime.

-----------------------------------------------------------------
-- DynamicOSI(tm) Release 1_2_1: RELEASE NOTES
-----------------------------------------------------------------

* ASN.1 <==> XML Translation API

  - Update: updated to handle invalid GenTime formats which
            included a trailing '.', '+', or '-' with no
            additional characters. Instead of throwing an
            exception, the API will now just return the core
            time structure.
-----------------------------------------------------------------
-- DynamicOSI(tm) Release 1_1_4: RELEASE NOTES
-----------------------------------------------------------------

* ASN.1 API

   - added new convenience methods which provide recursive
     access to complex nested data structures. These methods
     allow the user to easily access values inside nested SET,
     SEQUENCE, CHOICE, SET OF and SEQUENCE OF structes. Nested
     structures are addressed using a 'dot-form' field name
     notation where the components may be field names, choice
     labels or set/sequence-of indexes. The new methods are:
```

```
AbstractData     AbstractData.rref(String name)
void             AbstractData.rset(String name, String avn_value)
void             AbstractData.rset(String name, long value)
```

An example of this usage is:

```
AbstractData filter = AbstractData.Create("CMISFilter");

filter.rset("and.0.item.equality","{ administrativeState, locked }");

filter.rset("and.1.or.0.item.equality.attributeId",    "locationName");
filter.rset("and.1.or.0.item.equality.attributeValue", "ATL");
filter.rset("and.1.or.1.item.equality.attributeId",    "locationName");
filter.rset("and.1.or.1.item.equality.attributeValue", "DFW");
```

-- DynamicOSI(tm) Release 1_1_3: RELEASE NOTES
---

* ASN.1 Metadata Generator (Asn1Gen

- updated ASN.1 compiler to automatically convert ANY fields to ANY DEFINED BY fields where possible. If the compiler encounters an ANY field in a SET/SEQUENCE where there are only 2 fields, it will automatically use the first field as the defined by FIELD provided that the ANY is not the first field.

  This will allow for automatic encoding and decoding of ANY values for older ASN.1 definitions which pre-dated the ANY DEFINED BY construct.

-- DynamicOSI(tm) Release 1_1_2: RELEASE NOTES

-- Monfox Logger API

- Feature: Added the 'logDateFormat' attribute to the XML 'logger' configuration form. The value should be a 'SimpleDateFormat' format string.

  For example:

```
<logger outfile="debug.log"
        level="detailed"
        logDate="true"
        logDateFormat="yyyy-MM-dd+hh:mm:ss.SSS"/>
```

-- DynamicOSI(tm) Release 1_1_1: RELEASE NOTES

-- RFC1006 Responder API

- Fix: Updated to correct problem in sending responding-AP-title and responding-AE-qualifier when acting in responder mode.

```
-----------------------------------------------------------------
-- DynamicOSI(tm) Release 1_1_0: RELEASE NOTES
-----------------------------------------------------------------

* Fix: TSAP.startListener() not thread safe.

       This has been corrected through proper synchronization

-----------------------------------------------------------------
-- DynamicOSI(tm) Release 1_0_3: RELEASE NOTES
-----------------------------------------------------------------

* Added documentation, updated examples

-----------------------------------------------------------------
-- DynamicOSI(tm) Release 1_0_2: RELEASE NOTES
-----------------------------------------------------------------

* Minor API change TConnection.Listener. Removed TPDU dependencies

-----------------------------------------------------------------
-- DynamicOSI(tm) Release 1_0_1: RELEASE NOTES
-----------------------------------------------------------------

* Initial Availablility of standalone OSI Transport Layer API
  over RFC1006.

-----------------------------------------------------------------
-- DynamicOSI(tm) Release 1_0_0: RELEASE NOTES
-----------------------------------------------------------------

   -- Initial release
```

DynamicOSI™ Transport Layer/RFC1006 Java API

Developer Guide

A guide to developing Java based OSI Transport Layer Communications Applications using the DynamicOSI™ Transport Layer API

monfox	Document Type	**Developer Guide**
	Software Release	**1.x**
	Document Version	**1.1**
	Date	**January 31st, 2011**

Copyright © 2000-2011, Monfox, LLC. ALL RIGHTS RESERVED

MxLWare DRV APIs for MxL751SM

User Guide

Revision 1.1.2.1

Nov 09, 2009

Information contained in this document is Company Private to
MaxLinear Inc. and shall not be used, copied,
reproduced, or disclosed in whole or in part without the
written consent of MaxLinear.

Exhibit J

OpenCL for OS X

OpenCL lets you tap into the parallel computing power of modern GPUs and multicore CPUs to accelerate compute-intensive tasks in your Mac apps. Use OpenCL to incorporate advanced numerical and data analytics features, perform cutting-edge image and media processing, and deliver accurate physics and AI simulation in games.

Reference, Guides, and Sample Code

Selecting a GPU for OpenCL on the Mac Pro (Late 2013)
Learn how to query the GPUs of the latest Mac Pro and select one or both for use by OpenCL.

OpenCL and OpenGL Support in OS X Mavericks
Get details on the OpenCL and OpenGL versions supported by Mac computers running OS X Mavericks.

OS X v10.9 API Diffs
Review the latest updates to the OpenCL.framework for OS X Mavericks.

OpenCL Programming Guide for Mac
Learn about OpenCL for OS X, including parallelizable routines, Grand Central Dispatch, memory objects, data sharing, and more.

OpenCL Specification
Get full details on the OpenCL standard from the OpenCL Specification version 1.2.

OpenCL N-Body Simulation Example
Uses OpenCL and OpenGL to simulate a classical N-Body gravitational system while efficiently using one or more GPUs.

OpenCL Hello World Example
View sample code on basic usage of OpenCL, calculating a mathematical square for a buffer of floating point values.

OpenCL Procedural Geometric Displacement Example
View sample code on binding OpenCL to existing OpenGL buffers.

OpenCL Procedural Grass and Terrain Example
View sample code on creating a procedural field of grass on a generated terrain model, which is then rendered with OpenGL.

Videos

Learn about OpenCL with video presentations and tutorials.

- Harnessing the Power of the Mac Pro with OpenGL and OpenCL
- Working with OpenCL
- What's New in OpenGL for OS X
- Core Image Effects and Techniques

Forums

Post questions and discuss OpenCL with other developers and Apple engineers.
View Forums

Using the OpenCL Logo

If your app uses OpenCL or you've created your own implementation, learn about licensing the OpenCL logo.

Tools and Related Resources

- OpenGL for OS X
- Xcode

OpenCL Hello World Exampl

Apple Inc. [US] https://developer.apple.com/library/mac/samplecode/OpenCL_Hell

Mac Developer Library

Developer

OpenCL Hello World Example

Table of Contents

Download Sample Code

About OpenCL Hello World Example

ReadMe.txt

hello.c

Revision History

Next

OpenCL Hello World Example

Last Revision:	Version 1.4, 2011-03-01 OpenCL kernel was missing void return qualifier (Full Revision History)
Build Requirements:	Mac OS X 10.6, Xcode 3.2
Runtime Requirements:	Mac OS X 10.6

A simple "Hello World" compute example showing basic usage of OpenCL which calculates the mathematical square (X[i] = pow(X[i],2)) for a buffer of floating point values.

Next

Copyright © 2011 Apple Inc. All Rights Reserved. Terms of Use | Privacy Policy | Updated: 2011-03-01

Feedback

Table of Contents

- Introduction
- Hello World!
- Basic Programming Sample
- OpenCL On OS X Basics
- Identifying Parallelizable Routines
- How the Kernel Interacts With Data in OS X OpenCL
- Using Grand Central Dispatch With OpenCL
- Memory Objects in OS X OpenCL
- Creating and Managing Image Objects In OpenCL
- Creating and Managing Buffer Objects In OpenCL
- Sharing Data Between OpenCL and OpenGL
- Controlling OpenCL / OpenGL Interoperation With GCD
- Using IOSurfaces With OpenCL
- Autovectorizer
- Tuning Performance On the GPU
- Improving Performance On the CPU
- Binary Compatibility Of OpenCL Kernels
- Revision History

Next

About OpenCL for OS X

OpenCL™ (Open Computing Language) is an open standard for cross-platform, programming of modern highly-parallel processor architectures. Introduced with OS X v10.6, OpenCL consists of a C99-based programming language designed for parallelism, a powerful scheduling API, and a flexible runtime that executes kernels on the CPU or GPU. OpenCL lets your application harness the computing power of these processors to improve performance and deliver new features based on compute-intensive algorithms.

In addition to support for the OpenCL 1.1 standard, OS X v10.7 adds integration between OpenCL, Grand Central Dispatch (*GCD*), and Xcode to make it even easier to use OpenCL in your application.

At a Glance

Using OpenCL is easier than ever as of OS X v10.7:

- OpenCL is fully supported by Xcode. The Xcode offline compiler removes a configuration step that used to have to be performed before the kernel could be run and facilitates debugging earlier in the development process. See Hello World!.
- You can write OpenCL functions in separate files and include them in your Xcode project. You can compile the kernels when your application is built, before it runs. This improves runtime performance.
- OpenCL now integrates with GCD, making it easier for you to focus on making your OpenCL kernels more efficient. See Using Grand Central Dispatch With OpenCL.
- The autovectorizer compiles and accelerates performance of kernels that run on the CPU up to four times without additional effort. The autovectorizer allows you to write one kernel that runs efficiently on both a CPU and a GPU. You can invoke the autovectorizer regardless of whether you are compiling from Xcode or building the kernels at runtime. Or you can disable the autovectorizer if necessary. See Autovectorizer.

You can, of course, continue to use code you've already written to the OpenCL 1.1 standard. But see Binary Compatibility Of OpenCL Kernels for a note about how to handle existing binaries.

Feedback