Khronos Group
Conformance Process

1. Change History

- **February 2007** Generic process document with Specification specifics contained in Attachments
- **July 2007** Added OpenKODE Attachment D, Adoption is made independent of membership
- **August 2007** Clarified that number of submissions is unlimited
- **V1 - Nov 2007** Corrected bug fix versioning numbering in Test Code Source Package, clarified OpenGL ES fees
- **V2 – Feb 2008** Updates to OpenKODE submission package in Attachment D
- **V3 – May 2008** Added OpenGL ES 2.0 in Attachment A
- **V4 – Oct 2008** Added OpenGL SC Attachment E, updated OpenGL ES Attachment A, added Submission Alterations, added ability to extend Review Period
- **V5 – May 2009** Clarified Review Committee terminology and process, moved press release wording to external trademark guidelines, added ability to request subcontractor anonymity, removed flow down from system to component conformance, updated OpenVG attachment to 1.1, inserted older OpenGL ES test process into Attachment AA, Added OpenCL 1.0 and
OpenSL ES 1.0 in Attachments F and G
- **V6 – Oct 2010** Added OpenMAX AL in Attachment H, COLLADA in Attachment I and OpenWF in Attachment J. Updated Attachment F to include OpenCL 1.1. Added Reference Submissions. Updated OpenSL ES Profile logos
- **V7 – Feb 2011** OpenCL 1.2 added to Attachment F
- **V8 - Jan 2013** Updated OpenGL ES Attachment to include OpenGL ES 3.0
- **V9 - Jul 2013** OpenGL 4.4 added to Attachment K
- **V10 - Mar 2014** Attachment A updated for OpenGL and OpenGL ES 3.1 Attachment F updated for FPGAs and OpenCL 2.0
- **V11 – Oct 2014** Updated Attachment A for OpenGL 4.5
Added Attachment K for OpenVX
- **V12 – Nov 2015** Added Attachment L for WebGL
Updated Attachment F for OpenCL 2.1
Updated all API logos
- **V13 – Feb 2016** Added Attachment M for Vulkan 1.0
OpenCL fee schedule clarified
- **V14 – Jul 2016** Corrected pricing in Attachment M for Vulkan 1.0
Added Pricing in Attachment K for OpenVX 1.1
Added Pricing in Attachment A for OpenGL ES 3.2
Added formal definition of Conformance Fees
- **V15 – Mar 2017** Added Pricing in Attachment E for OpenGL SC 2.0
Added Pricing in Attachment L for WebGL 2.0
Updated Attachment A for OpenGL and OpenGL ES

- **V16a – Nov 2017** Added pricing in Attachment K for OpenVX 1.2
  Fixed link for WebGL Conformance
  Updated logos and sizing
  Improve clarity around OpenGL 3.2-4.3 submissions
  Added Pricing in Attachment A for OpenGL 4.6
  Added Reference Submission to Glossary
- **V17 – Jan 2018** Updated Attachment F for OpenCL
- **V18 – Mar 2018** Updated Attachment M for Vulkan 1.0 and 1.1
- **V19 – Jun 2018** Added Attachment N for SYCL 1.2.1
- **V20 – Aug 2019** Added Attachment O for OpenXR 1.0
- **V21 – Aug 2019** Added pricing in Attachment K for OpenVX 1.3
- **V22 – Sep 2019** Updated Attachment M for Vulkan 1.2
- **V23 – Jul 2020** Updated Attachment O for OpenXR 1.0 Release
  Updated Attachment F for OpenCL 3.0
  OpenGL and OpenCL Layered Implementations
  Refreshed all current logos
- **V24 – Sep 2020** Updated Attachment F for OpenCL 3.0 Final
- **V25 – Aug 2021** Updated Attachment E for OpenGL SC
- **V26 – Dec 2021** Changed Attachment M Vulkan pricing to subscription model
  Updated Attachment F for OpenCL
- **V27 – Feb 2022** Added Attachment P for Vulkan SC 1.0
  Updated OpenGL / OpenGL ES Review Period
  Clarified (not changed) Vulkan Subscription pricing table
- **V28 – Jul 2023** Updated Attachment M for Vulkan Layered Implementations
- **V29 – Nov 2023** Update Attachment N for SYCL 2020

2. Definitions

“**Adopters**” means entities that have executed the Adopters Agreement for the Specification regardless of whether that party is a current Promoter or Contributor of Khronos.

“**Adopters Agreement**” means the Khronos contract to provide a) access to the Adopters Package. b) the right to make Submissions under the Process. and c) a license to use the Marks for Conformant Products.

“**Adopters Mailing List**” means a mailing list that is subscribed to by all current Adopters for the Specification and used for notices to Adopters and discussion threads related to conformance.

“**Adopters Package**” means the complete set of source code and other materials received by Adopters to enable them to follow the Process.

“**Associated Working Group**” means the Khronos working group controlling the specification being tested in a Submission.

“Board” means the Khronos Board of Promoters.

“**Conformant Implementation**” means an Implementation with a Successful Submission.

“Conformance Fee” means the applicable fee(s) that all Adopters pay to access the Adopters Package and to submit Submission Packages to the Submission Repository for applicable version(s) of a Specification.

“**Conformant Product Criteria**” means the criteria by which products may be associated with a Conformant Implementation as defined in the Specification Attachment.

“**Conformant Products**” means the products identified in the Submission that comply with the Conformant Product Criteria and so may use the Marks.

“**Expiration Date**” means, for Specifications that have a subscription-based Conformance Fee model, the date upon which the Adopter’s subscription term expires.

“**Implementation**” means the specific implementation of the Specification being tested.

“**Marks**” means the trademarks associated with the Specification as defined in the Specification Attachment.

“**Paid Specification Version**” means (i) for Specifications with a one-off Conformance Fee structure, the version of the Specification for which an Adopter has executed an Adopters Agreement and paid the Conformance Fee or (ii) for Specifications that have a subscription-based Conformance Fee structure, all published versions for a Specification to which the Adopter has a current subscription.

“Passing Criteria” means the criteria by which an Implementation is deemed to have successfully passed the Tests – as defined in the Specification Attachment. Khronos may change the Passing Criteria at any time, but such changes shall not invalidate previously Successful Submissions or Submissions in their Review Period.

“**Process**” means the process defined in this document by which products may be tested for conformance with the Specification.

“**Process Document**” means this document that defines the Khronos Conformance Process.

“Reference Submission” simply refers to an existing Successful Submission to identify additional Conformant Products that can legitimately use the same testing. This should be used instead of a Submission Update when the Successful Submission and Reference Submission are created by different Adopters.

“Review Notice” means a written notice sent to an Adopter to identity any issues related to a Submission.

“**Review Period**” means the period defined in the Specification Attachment during which Submissions are subject to peer review.

“**Reviewers**” means any Adopters plus any Khronos Promoters and Contributors who have signed the Khronos Conformance Test Source License Agreement.

“**Specification**” means the Khronos specification or specifications identified in an executed Adopters Agreement.

“Specification Attachment” means the Attachment to this Process Document that defines the process details specific to the Specification.

“**Submission**” means a complete set of results created by performing the Tests on an Implementation according to the Process and which are passed to Khronos.

“**Submission Alteration**” means an alteration to a Successful Submission passed to Khronos to correct factual errors discovered in the Submission.

“**Submission Correction**” means a correction to a Submission or Submission Update currently in its Review Period.

“**Submission Update**” means an update to a Successful Submission passed to Khronos to add additional Conformant Products to the Submission.

“**Submission Package**” means the data to be included in a Submission as defined in the Specification Attachment.

“Submission Repository” means the Khronos online upload area for Submissions.

“**Successful Submission**” means a Submission that has followed the Process and is deemed to pass the Tests.

“**Tests**” means the Conformance Tests created by Khronos for the Specification.

“Waiver Statement” means a written identification and description of a potential bug in the Tests and the source code change used to fix the bug.

3. Purpose

Khronos has created Tests and associated Process for the Specification to promote consistent multi-vendor implementations and to create an objective definition of conformance for the Specification so that only Conformant Products may use the Marks.

4. Conformance Fees

All Adopters pay the Conformance Fee(s) as defined in the Specification Attachment on execution of the Adopters Agreement. Access to the Adopters Package for a Specification shall not be provided unless the Conformance Fees for that Specification have been received by Khronos. No refunds shall be provided for Conformance Fees under any circumstances. If Conformance Fees are changed, those changes will not be charged retrospectively.

Except as otherwise set forth in the applicable Specification Attachment, executing the Adopters Agreement sets the Adopters Paid Specification Version at the version of Specification indicated on the signature page of the Adopters Agreement and there is no limit on the number of Submissions an Adopter may make for Implementations that implement any version of the Specification up to and including the Paid Specification Version. Khronos may offer discounted Conformance Fees for Khronos members or existing Adopters to upgrade their Paid Specification Version from a previous level, or to extend their subscription to an applicable Specification.

When a new major release of the Specification is released by Khronos the Adopters Package will be updated with the Tests for the new Specification version. Adopters may continue to use the Adopters Package for Submissions up to their Paid Specification Version. Use of the Adopters Package for versions beyond the Paid Specification Version may require the execution of a new version of the Adopters Agreement and payment of additional Conformance Fees depending on the nature and content of the update. In general, bug fixes and minor updates will typically be covered by the current Adopters Agreement and will not incur additional Conformance Fees.

Adopters are provided access to the Adopters Package on a password protected section of the Khronos website and are enabled to make an unlimited number of Submissions for any number of Implementations using any version of the Specification up to the Paid Specification Version.

Notwithstanding the above, certain Specification Attachments may describe a subscription-based Conformance Fee structure. For those Specifications, Adopters may access and make unlimited Specifications for any version of the Specification up and until the applicable Expiration Date, after which they may no longer make Submissions for that Specification.

If a party wishes to gain access to the Adopters Package solely to act as a subcontractor to another Adopter without any rights to make a Submission on its own behalf, then that party must sign the Adopters Agreement, but the Conformance Fees may be waived if the contracting Adopter and contracted Adopter both sign the Subcontractor Warranty contained in Attachment B of the Adopters Agreement and the waiver of fees is approved by the Board. The contracting Adopter may request to the Board to keep the identity of the contracted Adopter confidential from other Khronos members, which permission shall be granted at the sole discretion of the Board.

5. Test and Process Updates

Khronos may update this Process, the Tests or other parts of the Adopters Package, including the Conformant Product Criteria, for subsequent Submissions at any time at its sole discretion. Such updates shall not invalidate previously Successful Submissions and Conformant Products, or Submissions, Submission Updates and Conformant Products identified in Submissions in their Review Period.

Khronos may provide pre-release versions of a new version of the Adopters Package to all Adopters for feedback at its discretion and will announce the posting of a new version of the Adopters Package on the Adopters Mailing list at least one week before its release.

Khronos will make all previously released versions of the Adopters Packages, Process Document and Specification Attachments for all released versions of the Specification available to Adopters. New Submissions may use any version of the Tests that has not been withdrawn but are strongly encouraged to use the latest version. The Adopters Package, Process Document and Specification Attachment used in a Submission must be, or have been, current at the same time.

6. Conformance Test Source and Porting

The source code format for the Tests is defined in the Specification Attachment. The Tests are provided as is and the Adopter is responsible for porting and running the Tests on the Implementation to generate the necessary information for a Submission. Khronos cannot provide any support for porting Tests. All modifications made to the Tests source code are licensed back to Khronos for its unrestricted use.

The Adopter should make no changes to the source code that disable or change the intended operation of any test unless the Adopter identifies a potential bug in a test. Source code changes to work around implementation limitations and bugs are not permitted. For each potential bug the Adopter identifies in the Tests, the Adopter must change the test source and submit a complete Waiver Statement in the relevant Submission Package.

Adopters are encouraged to submit suggested changes to source code and header files that would potentially increase portability to the appropriate repository.

7. Submissions

Once the Tests run on the Implementation, and satisfy the Passing Criteria, an Adopter may create a Submission and upload it to the Submission Repository.

Khronos shall distribute an email to the Specification Working Group mailing list and the Adopters mailing list when a Submission is made to start the Review Period for that Submission.

A Submission shall contain a Submission Package which contains the information below with additions and subtractions from the information below as defined in the API attachment:

- The version of the Specification being tested.
- The release date of the Tarball, Tests, Process Document and Specification Attachment to this document that were used.
- A statement of conformance in which the Adopter lists Conformant Products that are certified by the Adopter to be covered by this Implementation.
- All bugs and associated Waiver Statements which should be loaded into Khronos Bugzilla bug-tracking system.
- All changed source – that should be checked into the Adopters Branch of the Khronos Subversion server or as part of the Submission Package as defined in the Specification Attachment.
- All make files used to build the conformance tests for the Implementation as defined in the Specification Attachment.
- Contact details at the Adopter for any questions or Review Notices for this Submission.

8. Reference Submissions

An Adopter may make a Reference Submission for a product that integrates a Conformant Implementation that is covered by an existing Submission. A Reference Submission contains:

- The number of a successful Submission for the integrated implementation. The referenced Submission may have been submitted by a different Adopter.
- A statement of conformance in which the Adopter lists additional Conformant Products that are certified by the Adopter to be covered by the referenced Submission.
- Contact details at the Adopter for any questions or Review Notices for this Submission.

A Reference Submission is considered a new Submission and may not be made after the Expiration Date (if applicable).

9. Submission Alterations, Updates, Corrections and Deletions

Successful Submissions and Reference Submissions may be altered using a Submission Alteration or updated using a Submission Update. All Submission Alterations and Submission Updates undergo the same review process as a Submission.

Submission Updates are used to add Conformant Products associated with a Conformant Implementation. Any updated Submissions shall continue to be available on the Submission Repository to maintain a complete record of Submission dates but marked as updated once a Submission Update is successful.

Submission Alterations are used to correct factual errors discovered in Successful Submissions. Proposed alterations may request deletion of a Successful Submission from the Submission Repository or request changes to be made to a Successful Submission and any related updated Submissions. A Submission Alteration must include the affected Submission numbers; all requested changes; and must be accompanied by a Submission Package if appropriate.

While a Submission, Submission Alteration or Submission Update is in its Review Period, the submitting Adopter may make a Submission Correction, replacing the original Submission, Submission Alteration or Submission Update and restarting the Review Period, or may delete the Submission, Submission Alteration or Submission Update entirely.

A Submission Update is considered a new Submission and may not be made after the Expiration Date (if applicable). Submission Alterations, Submission Corrections, and Submission Deletions are not considered new Submissions and can be made at any time.

10. Submission Review Process

All Submissions, Reference Submissions, Submission Alterations and Submission Updates are available for inspection by all Reviewers for the Review Period during which time any Reviewer may raise a Review Notice for any issues related with a Submission including but not limited to:

- Submission does not satisfy the Passing Criteria.
- An invalid combination of Tests, Process Document and Specification Attachment.
- Inappropriate source changes to the Tests.
- Incorrect reporting of results.
- Inappropriate promotion of back doors and non-conformance that are not in the best interest of the Specification’s standing in the industry or against the spirit of the Process.
- Bugs in the Tests that materially affect the success of a Submission.
- Inappropriate association of Conformant Products with an Implementation.

On the receipt of a Review Notice, or on the Agreement of the Associated Working Group, the Associated Working Group shall appoint a Review Committee consisting of at least five Reviewers other than the submitting Adopter to resolve any Review Notices as follows:

- A designated Reviewer should make direct contact with a submitting Adopter and make all reasonable efforts to clearly identify any concerns.
- If changes are agreed, the submitting Adopter may make a Submission Correction to correct the Review Notice.
- If Reviewer and Adopter do not agree on the need for a Submission Correction then the Review Committee shall take all available information under consideration and determine by majority vote (50% or more of the Review Committee members) whether the Submission needs a Correction.
- The Review Committee may extend a Review Period if the submitting Adopter does not respond to any concerns in a timely way or the Review Committee needs more time to resolve The Review Notice. The Review Committee must notify the Board immediately of a Review Period extension and the Board may decide to adjust the extension period in its sole discretion.
- If the Submitter feels that the Review Notice is being incorrectly applied it may request the Board to make a final decision, through its normal voting process, on the validity of the Submission. In this case the Review Period will be extended until the Board reaches a decision.
- If the Review Committee does not resolve a Review Notice or extend the Review Period within a Review Period then the Submission, Submission Update or Submission Alteration shall be deemed successful.

A Review Committee may recommend to the Associated Working Group to waive the remainder of a Review Period if there are no outstanding issues relating to a Submission or Submission Update. A Successful Submission enables the Implementation to be deemed to be a Conformant Implementation and any Conformant Products may use the Marks.

11. Conformant Products

A Conformant Implementation may demonstrate conformance for a number of Conformant Products that fall within the Conformant Product Criteria in the Specification Attachment.

Submissions for test and prototype Implementations are acceptable if an Adopter wishes to demonstrate conformance and use the trademark for that Implementation.

12. External Feedback

Khronos shall create a public forum and encourage any Promoter, Contributor, Adopter or external entity to make suggestions to Khronos on how the Tests and Process may be improved and identify issues that are negatively affecting the effectiveness of Tests.

All received comments shall be passed to Khronos and the Associated Working Group that shall take any appropriate action, entirely at their own discretion.

13. Trademark Guidelines

Adopters shall use the Marks in accordance with the guidelines provided in each Specification Attachment together with the current standard trademark use guidelines issued by Khronos, and currently located at: http://www.khronos.org/trademark_guidelines.

Additional guidelines and reference artwork are available here: https://www.khronos.org/news/logos.

Attachment A
Open Source OpenGL ES and OpenGL Conformance Process Details

A1. Change History and Version

- January 23, 2017 – first version
- October 2017 – Improve clarity around OpenGL 3.2-4.5 submissions, OpenGL 4.6 added
- July 2020 - Extended Conformant Product Criteria for layered implementations

A2. Paid Specification Versions Covered

All versions of OpenGL ES including: 1.1, 2.0 and 3.1-3.2, and versions of OpenGL from 3.2-4.6.

This Attachment covers process details for all OpenGL ES tests and OpenGL 4.6 tests released into open source after 23 January 2017. Specific process details for older versions of the tests are covered in other attachments:

- For closed source tests for OpenGL ES 1.1, 2.0 and 3.0-3.2 and OpenGL 3.2-4.5, released between 5 November 2008 and 23 January 2017 - see Attachment AA.

- For OpenGL ES 1.1 tests released prior to 5 November 2008 – see Attachment AB.

Adopters are always encouraged to use the most recent version of the tests if possible.

A3. Conformance Fees

Conformance Fees for each version of the Specification are given in the table below. Fees for a given version of a Specification enable submissions for all available previous versions and ongoing minor updates to that API. Adopters that are not Khronos members pay the Non-member Base Fee, Adopters who are members pay the Member Base Fee. Adopters who are upgrading from an older version of OpenGL ES may be eligible for an additional discount, as shown in the table:

OpenGL and OpenGL ES Adopter Conformance Fees			
	Member Base Fee	**Non-member Base Fee**	**Upgrade Fee**
OpenGL ES 1.1	$10K	$15K	From 1.0: Base Fee minus paid 1.0 Fees
OpenGL ES 1.1-2.0	$14K	$19K	From 1.0: Base Fee minus paid 1.0 Fees From 1.1: Base Fee minus $6K
OpenGL ES 1.1-3.0	$25K	$30K	From 2.0: Base Fee minus $5K
OpenGL ES 1.1-3.2	$25K	$30K	From 3.0: Base Fee minus $5K
OpenGL 3.2-4.5	$25K	$30K	No upgrade discount from previous versions
OpenGL 3.2-4.6	$45K	$60K	No upgrade discount from previous versions

A4. Test Source Code Packaging

The OpenGL and OpenGL ES Conformance Test Suite (CTS) is revised periodically to improve coverage and to fix bugs. Official releases are identified by version strings of the form *opengl-cts-X.Y.Z* and *opengl-es-cts-X.Y.Z,* where *X.Y* is the version of the OpenGL or OpenGL ES that the test applies to, and *Z* is the revision number of the test. Revision numbers are sequential integers starting at zero.

Adopters are encouraged to use the latest revision of the CTS, but may make submissions using an older version, unless the older version has been withdrawn. Khronos reserves the right to withdraw older versions and to reject new submissions using those versions, provided that a) a newer version of the test has been available for at least

180 calendar days, and b) notice of intent to withdraw the older version was published at least 180 days before the effective date.

Notice of intent to withdraw any test version, or of any change to the conformance test process, will be distributed by email to the Adopters email list. A list of currently accepted test versions, together with dated notice of any intent to withdraw any of those versions, is available at https://github.com/KhronosGroup/VK-GL-CTS/wiki.

The source code for the Conformance Tests is maintained as an open source project at https://github.com/KhronosGroup/VK-GL-CTS. Official releases are represented by tagged commits to that repository, using the version string as the tag. To obtain a copy of the source code for any released version of the test, install `git` and execute the command:

```
$ git clone https://github.com/KhronosGroup/VK-GL-CTS.git -b <version>
```

where `<version>` is the version string associated with the desired release. Versions of the source that do not carry a release tag may not be used to make conformance submissions.

A5. README file

All versions of the OpenGL and OpenGL ES conformance test contain a top-level file named README.md, referred to in this document as "the README file". The README file defines Submission Package format and Passing Criteria for the version in question and provides instructions for porting the tests and running them to generate conformance results.

A6. Submission Package

The format and contents of a valid Submission Package are defined in the SUBMISSION PACKAGE section of the README file contained in the source code distribution.

A7. Passing Criteria

Tests must be run as described in the README file included in the source code distribution, and the results must satisfy the requirements given in that file.

For OpenGL, OpenGL ES 2.0 and OpenGL ES 3.X, directions for running the test are contained in the section titled "Running the Tests", and requirements for passing are given in the section titled "Understanding the Results".

A8. Review Period

The review period for ordinary Submissions is 30 calendar days. The working group may approve shorter review periods in special situations, such as prior to a new specification release.

The review period for Submission Updates and Reference Submissions is 7 calendar days.

The working group may, at its sole discretion, vote to waive the balance of the Review Period and declare a Submission conformant at any point during the Review Period.

A9. Conformant Product Criteria

Implementations claimed as Conformant Products must be similar to the Conformant Implementation in the following ways:

(i) For non-layered implementations, the identical rendering pipeline – i.e., identical binaries and/or accelerator data path to the display, or if a JIT compiler is used to generate binaries, then the use of the identical JIT compiler binary, or new versions of the binaries and/or accelerator data path or JIT compiler binary that do not cause any previously passing test to fail.

(ii) For layered implementations over a certified DX12, conformant Vulkan or production Metal API: the same implementation software (i.e. built from the same source) and the same version of the underlying API including minor revisions to either that do not cause any previously passing test to fail, and for each OS, there must be Successful Submissions using at least two (if available) independent implementations of the underlying API from different vendors (if available), at least one of which must be hardware accelerated.

(iii) The same major version of the same OS that uses substantially similar display functionality, or minor version updates to the OS that do not cause any previously passing test to fail.

(iv) The identical set, or a subset, of supported configs. Different display resolutions are permitted.

Conformance may be claimed for any component-level products necessary for a Conformant Implementation.

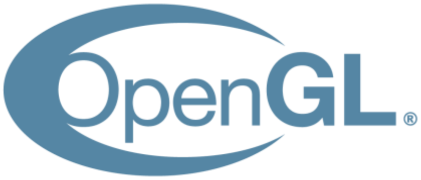
A10. Mark and Logo Usage Guidelines

The following usage guidelines must be followed for any use of the Marks identified below:

- "®" must be used as shown with the first use of the written Mark in a document.
- Each document that uses the Marks must include the following text: "OpenGL, OpenGL ES, and the OpenGL logo are registered trademarks, and the OpenGL ES logo is a trademark of Silicon Graphics International, used by permission by Khronos."

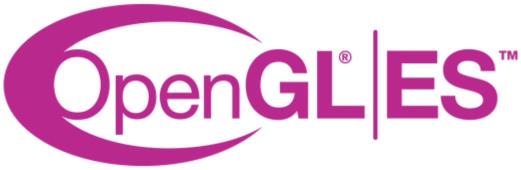
The Marks for OpenGL are:

- OpenGL®
- The OpenGL Logo:



The Marks for OpenGL ES are:

- OpenGL® ES
- The OpenGL ES Logo:



Attachment AA
Closed Source OpenGL ES and OpenGL Conformance Process Details

AA1. Change History and Version

- **February 2007 –** first version attached to generic process document
- **May 2008** – added process for OpenGL ES 2.X conformance testing. restructure and clarified Conformance Fees and source code packaging
- **October 2008** – updated process for OpenGL ES 1.X conformance testing for compatibility with version 2 of the test
- **January 2009** – added reference to Attachment AA for older version of the tests
- **January 2013** – update for OpenGL ES 3.0
- **July 2013 –** update for OpenGL 4.4
- **March 2014** – OpenGL process merged, added OpenGL ES 3.1
- **October 2015** – update for OpenGL 4.5
- **July 2016 –** added pricing for OpenGL ES 3.2
- **October 2017** – clarification for OpenGL 3.2 through OpenGL 4.6

AA2. Paid Specification Versions Covered

This Attachment covers only the conformance process specifics for the closed source tests for OpenGL ES 1.1, 2.0 and 3.0-3.2 and OpenGL 3.2-4.5, released between 5 November 2008 and 23 January 2017.

See Attachment A for more information, other test versions, and pricing information.

Adopters are always encouraged to use the most recent version of the tests if possible.

AA3. Conformance Fees

As per Attachment A.

AA4 Test Source Code Packaging

OpenGL Conformance Tests will automatically detect the version supported by the Implementation and restrict tests to those appropriate for that version. Conformance test processes for OpenGL ES 1.X differ from those for the other APIs. Each section of this Attachment describes both processes.

The source code for the Conformance Tests is packaged in a gnu-zipped tar file. The file name has the format: GLCTS-<X.Y>.<Z>.<W> -<year><month><day>.tgz for OpenGL and ESCTS-<X.Y>.<Z>.<W> -<year><month><day>.tgz for OpenGL ES. Here <X.Y> is the version of the OpenGL or OpenGL ES Specification to which the test applies, <Z> is the major revision number of the conformance test, and <W> is the minor revision number of the test. Changes in minor revision number reflect changes intended to correct bugs or improve portability and maintainability of the tests. Changes in major revision number reflect changes that significantly expand test coverage and/or impose stricter Passing Criteria. For example, a test labeled ESCTS-2.0.1.0-20080512.tgz applies to OpenGL ES 2.0, and is version 1.0 of the conformance test, released on May 12th, 2008.

When a new test release is created, a tag is added to the conformance test Subversion (SVN) repository so that Adopters who wish to can sync the SVN tree against a particular bugfix release. The SVN tag for a release has the same name as the .tgz file, without the .tgz extension.

As specified in Section 5 of the Conformance Process Document, Adopters may make Submissions using any version of the Tests and Process but are encouraged to use the most recent version that their implementations are able to pass.

All versions of the Conformance Tests for OpenGL, OpenGL ES 3.X or OpenGL ES 2.X, and all versions of the OpenGL ES 1.X source code tree except the first, contain a top-level file referred to in this document as “the README file”. Those files define Submission Package format and Passing Criteria and provide detailed instructions for porting the tests and running them to generate conformance results. The file in the OpenGL and OpenGL ES 3.X source trees is named `GL_and_ES3_Readme.pdf`, and the file in the OpenGL ES 2.X source tree is named `ES2_Readme.pdf`. These files are in Portable Document Format. The file in the OpenGL ES 1.X source tree is an ASCII text file named `README`.

AA5. Submission Package

The format and contents of a valid Submission Package are defined in the SUBMISSION PACKAGE section of the README file contained in the source code distribution.

AA6. Passing Criteria

Tests must be run as described in the README file included in the source code distribution, and the results must satisfy the requirements given in that file.

For OpenGL ES 1.0 and 1.1, directions for running the test are given in the EXECUTION section of the README file, and requirements for passing the test are given in the PASSING CRITERIA section.

For OpenGL, OpenGL ES 2.0 and OpenGL ES 3.X, directions for running the test are contained in the section titled “Running the Tests”, and requirements for passing are given in the section titled “Understanding the Results”.

AA7. Review Period

As per Attachment A.

AA8. Conformant Product Criteria

As per Attachment A.

AA9. Marks and Usage Guidelines

As per Attachment A.

Attachment AB
Older OpenGL ES 1.1
Conformance Process Details

AB1. Change History and Version

- **January 2009** – Inserted to cover older of OpenGL ES 1.1 tests

AB2. Paid Specification Versions Covered

This Attachment covers only conformance process specifics for the older version of the OpenGL ES 1.1 conformance tests released prior to 5 November 2008.

See Attachment A for more information, other test versions, and pricing information.

Adopters are always encouraged to use the most recent version of the tests if possible.

AB3. Conformance Fees

As per attachment A.

AB4. Test Source Code Packaging

The source code for the Tests is packaged in a gnu-zipped tar file. It is named using the following format: conformes-<OpenGL-ES version>r<revision number>-<year><month><day>.tgz. The revision number is incremented for every bug fix release of the Tests targeted at a specific version of the Specification. When the tar file is made, a tag is added to the conformance test SVN repository so that Adopters can sync the SVN tree against a particular bugfix release. The SVN tags use the following format: OGLES-<major version>-<minor version>-rev<version number>.

AB5. Submission Package

A Submission must contain the information defined in the Submission section of the process document PLUS all of the following Specification specific information:

- Identification of the Implementation including: the CPU running the Specification, the OS, the Specification pipeline and display configuration – including version numbers.
- The result log (output) for the executed tests, run in the order and with the parameters specified in the "EXECUTION" section of the "README" file located at the top level of the conformance test source tree.
- The complete source of the executed tests together with an annotated diff file containing any source changes packaged as a ZIP archive file (.zip) or gnuzip compressed tarball (.tar.gz) rooted at the top level of the conformance test source tree in the same way as the distributed tarball, with all generated files such as objects and libraries removed, and with a README-<company name> at the top summarizing the changed files. The annotations should make it clear what changes have been made and for what reason.
- The result log, README-<company name>, and statement of conformance must each be plain text files readable in a simple text editor.

A Submission for an Implementation of the Common Profile must contain two result logs and all necessary source updates following execution of the Tests on both the Common and Common-Lite libraries. A Submission for an Implementation of the Common-Lite Profile need only contain a single result log and source updates following execution of the Tests on the Common-Lite library.

AB6. Passing Criteria

A conforming config must unconditionally pass the “covgl”, “covegl”, and “primtest” tests as well as all tests in the "Must Pass" test group of “conform” and may fail no more than 7 (seven) “conform” tests in any other test group at any single path level.

A Conformant Implementation must: include at least one conformant config. the ratio of conformant configs to non-conformant configs must be equal to or greater than 1 (one). and there must be one or more conformant configs with a Z buffer of at least 15-bit depth.

AB7. Review Period

As per Attachment A.

AB8. Conformant Product Criteria

Conformant Products must be similar to the Conformant Implementation in the following ways:

- the identical Specification pipeline – i.e., identical binaries and/or accelerator data path to the display, or if a JIT compiler is used to generate binaries, then the use of the identical JIT compiler binary, or new versions of the binaries and/or accelerator data path or JIT compiler binary that do not cause any previously passing test to fail.
- the same major version of the same OS that uses substantially similar display functionality or minor version updates to the OS that do not cause any previously passing test to fail.
- the identical set, or a subset, of supported configs. Different display resolutions are permitted.

AB9. Marks and Usage Guidelines

As per Attachment A.

Attachment B
OpenMAX IL Conformance Process Details

B1. Change History and Version

- **February 2007** – first release

B2. Paid Specification Versions Covered

Up to OpenMAX IL 1.1 including minor release updates (i.e., OpenMAX IL 1.1.1 / 1.1.2 etc. are covered).

B3. Conformance Fees

The Conformance Fee for each version of the Specification is below – for Adopters that are not members of Khronos – these fees shall be increased by $5K:

	Conformance Fee	**Upgrade Conformance Fee**
Any version of OpenMAX up to OpenMAX IL 1.1	$10K	NA

B4. Tests Source Code Packaging

The source code for the Tests is packaged in a gnu-zipped tar file. It is named using the following format: conformes-<OpenMAX-IL version>r<revision number>-<year><month><day>.tgz. The revision number is incremented for every bug fix release of the Tests targeted at a specific version of the Specification. When the tar file is made, a tag is added to the conformance test SVN repository so that Adopters can sync the SVN tree against a particular bugfix release. The SVN tags use the following format: OMXIL-<major version>-<minor version>-rev<version number>.

B5. Submission Package

A Submission must contain the information defined in the Submission section of the process document PLUS all of the following Specification specific information:

- Identification of the Implementation including: the CPU running the Specification, the OS and the Specification pipeline – including version numbers.
- The result log (output) for the executed tests, run in the order and with the parameters specified in the "EXECUTION" section of the "README" file located at the top level of the conformance test source tree.
- The complete source of the executed tests together with an annotated diff file containing any source changes packaged as a ZIP archive file (.zip) or gnuzip compressed tarball (.tar.gz) rooted at the top level of the conformance test source tree in the same way as the distributed tarball, with all generated files such as objects and libraries removed, and with a README-<company name> at the top summarizing the changed files. The annotations should make it clear what changes have been made and for what reason.
- The result log, README-<company name>, and statement of conformance must each be plain text files readable in a simple text editor.

B6. Passing Criteria

A conforming config must unconditionally pass the “base profile” tests as well as all tests in the "standard component" test group representative of claimed standard roles (if any). A conforming config claiming “interop” must also unconditionally pass the “interop profile” tests.

B7. Review Period

30 Days

B8. Conformant Product Criteria

Conformant Products must be similar to the Conformant Implementation in the following ways:

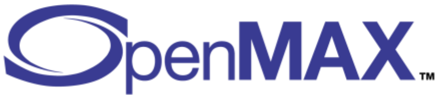
- the identical Specification implementation, i.e., identical binaries and/or accelerator data path to the component output, or new versions of the binaries and/or accelerator data path that do not cause any previously passing test to fail.
- the same major version of the same OS that uses substantially similar media processing functionality, or minor version updates to the OS that do not cause any previously passing test to fail.

B9. Marks and Usage Guidelines

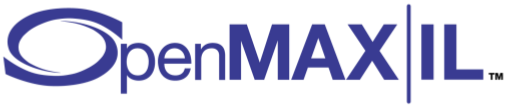
The following usage guidelines must be followed for any use of the Marks below:
- "™" must be used as shown with the first use of the written Mark in a document.
- the following text must be included in each document that uses the Marks: "OpenMAX and the OpenMAX logo are trademarks of the Khronos Group Inc."

(i) OpenMAX™ IL

(ii) OpenMAX Logo:



(iii) OpenMAX IL Logo:



Attachment C
OpenVG Conformance Process Details

C1. Change History and Version

- **February 2007 –** first release
- **January 2009 –** updated for OpenVG 1.1

C2. Paid Specification Versions Covered

OpenVG 1.0 and OpenVG 1.1 including minor release updates (i.e., OpenVG 1.0.1 / 1.0.2, OpenVG 1.1.1 / 1.1.2 etc. are covered).

C3. Conformance Fees

The Conformance Fee for each version of the Specification is below – for Adopters that are not members of Khronos – these fees shall be increased by $5K:

	Conformance Fee	**Upgrade Conformance Fee**
OpenVG 1.0 and OpenVG 1.1	$10K	$5K

C4. Tests Source Code Packaging

The source code for the Tests is packaged in a gnu-zipped tar file. It is named using the following format: conformvg-<OpenVG version>r<revision number>-<year><month><day>.tgz. The revision number is incremented for every bug fix release of the Tests targeted at a specific version of the API. When the tar file is made, a tag is added to the conformance test SVN repository so that Adopters can sync the SVN tree against a particular bugfix release. The SVN tags use the following format: OVG-<major version>-<minor version>-rev<version number>.

C5. Submission Package

A Submission must contain the information defined in the Submission section of the process document PLUS all of the following Specification specific information:

- The version of the API being tested and the revision and date of the conformance tests that were used, the CPU running the API, the OS, the API pipeline and display configuration, and supported surface formats and color spaces – including version numbers - used in the Implementation
- The result log (output) for the executed tests, run in the order and with the parameters specified in the "EXECUTION" section of the "README" file located at the top level of the conformance test source tree.
- The complete set of results (images) produced by executed tests.
- The complete source of the executed tests together with an annotated diff file containing any source changes Adopters Packaged as a ZIP archive file (.zip) or gnuzip compressed tarball (.tar.gz) rooted at the top level of the conformance test source tree in the same way as the distributed tarball, with all generated files such as objects and libraries removed, and with a README-<company name> at the top summarizing the changed files. The annotations should make it clear what changes have been made and for what reason.
- All make files used to build the conformance tests for the Implementation.
- The result log, README-<company name>, and statement of conformance must each be plain text files readable in a simple text editor.

C6. Passing Criteria

The conformance test suite consists of over 300 test cases, grouped into 12 groups of tests. Conformant implementation must pass all test cases.

Definitions

- A VG-Supporting EGL config is one whose EGL_RENDERABLE_TYPE attribute contains EGL_OPENVG_BIT.
- A VG-Conformant config is a VG-Supporting config whose EGL_CONFORMANT attribute contains EGL_OPENVG_BIT.
- A VG-Nonconformant config is a VG-Supporting config which is not VG-Conformant.

OpenVG conformance has been designed to be a 'Must Pass' procedure. For each VG-Conformant EGL config, the tests will be run against each valid combination of EGL_VG_COLORSPACE and EGL_VG_ALPHA_FORMAT allowed by the EGL_SURFACE_TYPE attribute of that config. All combinations must unconditionally pass all tests in order for the config to pass conformance.

A Conformant Implementation must include at least one VG-Conformant EGL config. All VG-Conformant configs must pass the conformance tests. Finally, the ratio of VG-Conformant configs to VG-Nonconformant configs must be equal to or greater than 1 (one).

For OpenVG implementations not exporting EGL, whatever alternate method the implementation uses to describe pixel formats (aka "non-EGL configs") must satisfy these properties:

- A non-EGL config must unconditionally pass all tests under all combinations of attributes defined by that config which affect the OpenVG implementation in order for that config to pass conformance.
- If a mechanism exists for a non-EGL config to describe whether or not it is conformant, and that mechanism claims the config is conformant, then the config must pass conformance.
- A Conformant implementation must include at least one conformant non-EGL config, and the ratio of configs passing conformance to configs not passing conformance must be greater than or equal to 1 (one).

C7. Review Period

30 Days

C8. Conformant Product Criteria

Conformant Products must be similar to the Conformant Implementation in the following ways:

- The identical Specification pipeline – i.e., identical binaries and/or accelerator data path to the display, or if a JIT compiler is used to generate binaries then the use of the identical JIT compiler binary, or new versions of the binaries and/or accelerator data path or JIT compiler binary that do not cause any previously passing test to fail.
- The same major version of the same OS that uses substantially similar display functionality or minor version updates to the OS that do not cause any previously passing test to fail.
- The identical set, or a subset, of supported configs. Different display resolutions are permitted.

C9. Marks and Usage Guidelines

The following usage guidelines must be followed for any use of the Marks below:
- "™" must be used as shown with the first use of the written Mark in a document
- the following text must be included in each document that uses the Marks: "OpenVG and the OpenVG logo are trademarks of the Khronos Group Inc."

(i) OpenVG™

(ii) OpenVG Logo:

Attachment D
OpenKODE Conformance Process Details

D1. Change History and Version

- **February 2008** – first release

D2. Paid Specification Versions Covered

OpenKODE 1.0 Provisional 1.0 (any revision) and OpenKODE 1.0 including minor release updates (i.e. OpenKODE 1.0.1 / 1.0.2 etc. are covered).

Note that when tests for finalized OpenKODE 1.0 are released, any tests for provisional versions of the OpenKODE specification will be invalidated – and may not be used in any Submissions for OpenKODE 1.0 or later.

D3. Conformance Fees

The Conformance Fee for each version of the Specification is below – for Adopters that are not members of Khronos – these fees shall be increased by $5K:

	Conformance Fee	**Upgrade Conformance Fee**
OpenKODE 1.0 Provisional (any revision) and OpenKODE 1.0	$10K	NA

D4. Tests Source Code Packaging

The source code for the Tests is packaged in a gnu-zipped tar file. It is named using the following format: conformkode-<OpenKODE version>r<revision number>-<year><month><day>.tgz. The revision number is incremented for every bug fix release of the Tests targeted at a specific version of OpenKODE. When the tar file is made, a tag is added to the conformance test SVN repository so that Adopters can sync the SVN tree against a particular bugfix release. The SVN tag of a particular revision of the Tests has the same name as the gnu-zipped tar file, without the .tgz extension.

D5. Submission Package

A Submission must contain the following information contained in the top-level directory of a zip file:

- Submission information in a file called “submission_details.txt” in the format defined in the OpenKODE Implementation Details Template document “submission_details_template.txt”.
- A file called “build_information.txt” listing any source files changed to fix bugs and summarizing the build procedure. Any detailed build information or make files are optional but build information for the Conformance Tests may be submitted to clarify the submission and reduce the chance for queries and delays.
- A folder containing the source of all tests edited to fix bugs, with all generated files such as objects and libraries removed, together with annotated diff files making it clear what changes have been made and for what reason.
- The submission_details.txt and build_information.txt files must each be plain text files readable in a simple text editor.
- All result logs for all executed tests as specified in the "EXECUTION" section of the "README" file located at the top level of the conformance test source tree.

D6. Passing Criteria

A Conformant Implementation must pass: all OpenKODE Core tests, all tests for all extensions supported by this implementation, and all relevant test cases appropriate to the included media APIs, with no changes to any Test source other than for compensating for bugs in the test or the specification. All media APIs included in the Implementation must have individual Successful Submissions at the time of the OpenKODE Submission.

D7. Review Period

30 Days

D8. Conformant Product Criteria

Conformant Products must be similar to the Conformant Implementation in the following ways:

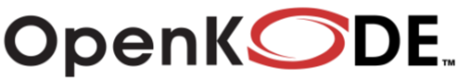
- Each Conformant Product must use the identical Specification binaries or new versions of the binaries that contain only bug fixes and no new OpenKODE functionality and that do not cause any previously passing test to fail.
- Each Conformant Product must use the same or similar (as defined by the individual media API Conformant Product Criteria) implementation of all submitted media APIs.
- Each submitted media API must be conformant on each Conformant Product (as defined by the individual media API Conformant Product Criteria).
- Each Conformant Product must use the same major and minor version of the same OS or platform (including UI framework if appropriate).
- Each Conformant Product must use no extra IO indexes that are in the Specification.
- Each Conformant Product must use the same CPU version or a different CPU version that has no impact on OpenKODE functionality.

D9. Marks and Usage Guidelines

The following usage guidelines must be followed for any use of the Marks below:
- "™" must be used as shown with the first use of the written Mark in a document
- the following text must be included in each document that uses the Marks: "OpenKODE and the OpenKODE logo are trademarks of the Khronos Group Inc."

(i) OpenKODE™

(ii) OpenKODE Logo:



The following usage guidelines can be used to optionally indicate which media APIs are included in the Conformant Implementation, note each API listed must have the appropriate ® or ™:

(i) OpenKODE™ (with OpenGL® ES, OpenVG™)

(ii) OpenKODE™ 1.0 (with OpenGL® ES 1.1, OpenVG™ 1.0)

Any further details of a Conformant Implementation as defined in the Submission Package section above may also be listed.

Attachment E
OpenGL SC Conformance Process Details

E1. Change History and Version

- **May09** – OpenGL SC 1.0 Adopters Program released
- **Apr16 –** added OpenGL SC 2.0
- **Aug21** - Moved Submission Package definition to source code package, clarified that OpenGL SC 2.0 Conformance Fees also covers OpenGL SC 1.0 submissions, update to Conformant Product Criteria for offline compilers

E2. Paid Specification Versions Covered

OpenGL SC 1.0 including minor release updates (i.e., OpenGL SC 1.0.1 / 1.0.2 etc. are covered).

OpenGL SC 2.0 including minor release updates

E3. Conformance Fees

	Khronos Member	**Non-member**
OpenGL SC 1.0 and minor specification updates	$20K	$25K
OpenGL SC 1.0, OpenGL SC 2.0 and minor specification updates	$30K	$40K

E4. Test Source Code Packaging

The source code for the Tests is packaged in a gnu-zipped tar file. It is named using the following format: conformsc-<OpenGL SC version>r<revision number>-<year><month><day>.tgz. The revision number is incremented for every bug fix release of the Tests targeted at a specific version of OpenGL SC. When the tar file is made, a tag is added to the conformance test SVN repository so that Adopters can sync the SVN tree against a particular bugfix release. The SVN tag of a particular revision of the Tests has the same name as the gnu-zipped tar file, without the .tgz extension.

E5. Submission Package

The format and contents of a valid Submission Package are defined in the SUBMISSION PACKAGE section of the README file contained in the source code distribution.

E6. Passing Criteria

A Conformant Implementation must pass: all tests, all tests for all extensions supported by this implementation with no changes to any Test source other than for compensating for bugs in the test or the specification.

E7. Review Period

30 Days

E8. Conformant Product Criteria

Conformant Products must be similar to the Conformant Implementation in the following ways:

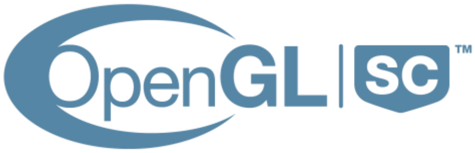
- the identical rendering pipeline – i.e., identical binaries and/or accelerator data path to the display, identical offline compiler binary, or new versions of the binaries and/or accelerator data path or offline compiler binary that do not cause any previously passing test to fail.
- the same major version of the same OS that uses substantially similar display functionality or minor version updates to the OS that do not cause any previously passing test to fail.
- the identical set, or a subset, of supported configs. Different display resolutions are permitted.

E9. Marks and Usage Guidelines

The following usage guidelines must be followed for any use of the Marks below:
- "®" must be used as shown with the first use of the written Mark in a document
- the following text must be included in each document that uses the Marks: "OpenGL is a registered trademark, and the OpenGL SC logo is a trademark of Silicon Graphics Inc. used by permission by Khronos."

(i) OpenGL® SC

(ii) OpenGL SC Logo:



Attachment F
OpenCL Conformance Process Details

F1. Change History and Version

- **May 2009** – OpenCL 1.0 Adopters Program released
- **May 2010** – OpenCL 1.1 Adopters Program added
- **March 2011** – OpenCL 1.2 Adopters Program added
- **March 2012** – OpenCL 2.0 Adopters Program added
- **October 2015 –** OpenCL 2.1 Adopters Program added and change to GitHub
- **February 2016** – Fee schedule clarified
- **October 2017** – OpenCL 2.2 Fees added
- **July 2020** – OpenCL 3.0 Fees, layered implementation Conformant Product Criteria
- **September 2020** – Tighten origin for CTS source; tidy pricing and passing criteria
- **December 2021** – Use of tagged versions, Passing Criteria clarified

F2. Paid Specification Versions Covered

All versions of OpenCL 1.X up to OpenCL 1.2 including maintenance updates, OR all versions of OpenCL 1.X, 2.X and 3.0, including maintenance updates. Paid fees do not cover specifications with unreleased tests at the time of payment, e.g.: if fees were paid for OpenCL 2.X before OpenCL 3.0 tests were released, OpenCL 3.0 fees from the table below must be paid to enable OpenCL 3.0 submissions.

F3. Conformance Fee Schedule

	Academic Member or Academic Adopter	**Khronos Member**	**Non Member**
OpenCL 1.0/1.1/1.2 and minor specification updates	$1,500	$10,000	$15,000
OpenCL 1.0/1.1/1.2 OpenCL 2.0/2.1/2.2 OpenCL 3.0 and minor specification updates	$1,500	$25,000	$30,000

F4. Test Source Code Packaging

The source code for the conformance test shall be taken from a tag in https://github.com/KhronosGroup/OpenCL-CTS.

A conformance submission is valid only if it is made from a tagged version of the conformance test suite, where the tagged commit date is within 6 months of the submission date. If any patches to the tagged version are necessary to fix test bugs, they must be cherry-picked from subsequent main branch commits and applied without merge conflicts. Any required patches must be documented in the submission as a list of commit SHAs.

F5. Submission Package

A Submission must contain the following information contained in the top-level directory of a zip file:

- Submission information in a file called “submission_details.txt” in the format defined in the OpenCL Implementation Details Template document “submission_details_template.txt”. The template can be found in the conformance package under the test_conformance directory.
- A file called “build_information.txt” that summarizes the build options (CMake options) and environment used (toolchain, etc).
- The submission_details.txt and build_information.txt files must each be plain text files readable in a simple text editor.
- All result logs for all executed tests.
- Platforms and Devices are handled in Submissions according to the following rules:
- Conformance can be claimed separately for a Platform or Device, but every Submission must contain a Platform and at least one Device.
- A Device is tested against the version of the tests for the claimed version of Conformance.
- If Conformance is claimed for a platform – it must have at least one conformant device at the claimed specification version.
- A Submission can contain multiple device test results by including a log file per device, together with a completed template for the entire submission.

F6. Passing Criteria

A Successful Submission must contain output from the required set of subtests, using the source code described in Section F. The required set of subtests is defined as every subtest that would be run by executing each command line with the arguments listed in opencl_conformance_tests_full.csv. The tests identified in the following paragraph may be run with the specified optimizations. A list of affected tests and the justifications for using the optimizations must be included in submission_details.txt.

If an OpenCL Device implements operations on vector types using corresponding operations on scalar types, it is sufficient to run conformance tests on the scalar operations and a subset of the vector operations across the vector lengths. As an example, if a Device implements all conversions on 2,3,4,8 and 16 vectors using conversions on the corresponding scalar type, the test_conversions CTS binary can be run with the -1 option to get coverage on scalars. The binary can then be run with the -w option for each of the vector variants. i.e., '-w -2', '-w -3', '-w -4', '-w -8' and '-w -16'.

The output must indicate that each test or subtest was passed and may not contain test errors or failures.

The specification version tested must be reported for the Platform and each device. Each OpenCL device may be tested at the same or lower version number than the Platform. The Platform and at least one device must be tested at the Version of the specification that conformance is claimed for.

A Submission for Implementations that would be Conformant Products under an existing Submission according to the criteria in section F8, except for the use of a different OS, may use the ‘wimpy’ mode of the test_conversions and bruteforce conformance sub-tests.

All tests must use the ICD if an ICD is present on the Conformant Product.

Platforms must run math brute force and conversion tests in normal mode using both OpenCL C and SPIR-V (if supported) unless the platform uses the materially equivalent underlying implementation, including built-in functions, for OpenCL C and SPIR-V, in which case those tests may be run in wimpy mode.

F7. Review Period

The default Review Period for Submissions, Submission Alterations and Submission Corrections is 30 calendar days. The default review period for Submission Updates and Reference Submissions is 7 calendar days.

The working group may approve shorter Review Periods for all Submissions, during a working group-agreed period, such as prior to a new specification release.

The working group may, at its sole discretion, vote to waive the balance of a Review Period to convert any type of Submission to a Successful Submission at any time.

F8. Conformant Product Criteria

Conformant Products may be OpenCL Devices or Platforms. Conformant Products must be similar to the Conformant Implementation in the following ways:

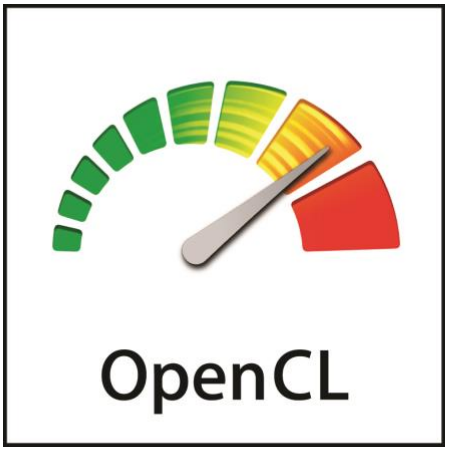
- For OpenCL Devices each Conformant Product must use:
    - For non-layered implementations on devices with an Instruction Set Architecture (ISA): the materially equivalent device ISA and binaries or version updates to binaries that do not cause any previously passing test to fail.
    - For non-layered implementations on devices without an Instruction Set Architecture (ISA) (e.g., FPGAs): a materially equivalent compilation flow that does not cause any previously passing test to fail.
    - The same major and minor version of OpenCL with the same set or a subset of supported extensions and enabled features.
    - For layered implementations over a certified DX12, conformant Vulkan or production Metal API: the same implementation software (i.e., built from the same source) and the same version of the underlying API including minor revisions to either that do not cause any previously passing test to fail.
    - A conformant OpenCL Platform at an OpenCL version equal to or higher than the platform used to test the Device provided that no previously passing tests fail.
- For OpenCL Platforms each Conformant Product must use:
    - The same major version of the same OS (or different flavors of Linux with the same major and minor version of the OS kernel) or minor version updates to that OS that do not cause any previously passing test to fail. or the equivalent driver model version or minor version updates to the driver model that do not cause any previously passing test to fail.
    - For layered implementations, for each OS, there must be Successful Submissions using at least two (if available) independent implementations of the underlying API from different vendors (if available), at least one of which must be hardware accelerated.
    - The materially equivalent host ISA.
    - The same major and minor version of OpenCL with the same set or a subset of supported extensions and enabled features.
    - The materially equivalent math library used while running the tests, or minor version updates to the math library that do not cause any previously passing test to fail.
    - The same OpenCL Device, or other Conformant OpenCL Devices that do not cause any previously passing test to fail.
    - The same OpenCL ICD, if present, or a later version of the ICD that does not cause any previously passing test to fail.

F9. Marks and Usage Guidelines

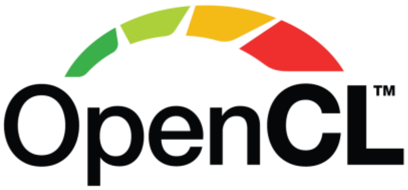
The following usage guidelines must be followed for any use of the Marks below:
- “TM” must be used as shown with the first use of the written Mark in a document
- the following text must be included in each document that uses the Marks: “OpenCL and the OpenCL logo are trademarks of Apple Inc. used by permission by Khronos.”

(i) OpenCL™

(ii) Original OpenCL Logo:



(iii) New OpenCL Logo (preferred):



Attachment G
OpenSL ES Conformance Process Details

G1. Change History and Version

- **May 2009** – OpenSL ES 1.0 Adopters Program released

G2. Paid Specification Versions Covered

OpenSL ES 1.0 including minor release updates (i.e., all OpenSL ES 1.x versions are covered).

G3. Conformance Fee

	Khronos Member	Non-member
OpenSL ES 1.0 and minor specification updates	$10K	$15K

G4. Tests Source Code Packaging

The source code for the Tests is packaged in a gnu-zipped tar file. It is named using the following format: conformes-<OpenSL-ES version>r<revision number>-<year><month><day>.tgz. The revision number is incremented for every bug fix release of the Tests targeted at a specific version of the Specification. When the tar file is made, a tag is added to the conformance test SVN repository so that Adopters can sync the SVN tree against a particular bugfix release. The SVN tags use the following format: OSLES-<major version>-<minor version>-rev<revision number>.

G5. Submission Package

A Submission must contain the information defined in the Submission section of the process document PLUS all of the following Specification specific information:

- Identification of the Implementation including: the CPU running the Specification, the OS and the Specification pipeline – including version numbers.
- The result log (output) for the executed tests, run in the order and with the parameters specified in the "EXECUTION" section of the "README" file located at the top level of the conformance test source tree.
- The complete source of the executed tests together with an annotated diff file containing any source changes packaged as a ZIP archive file (.zip) or gnu zip compressed tarball (.tar.gz) rooted at the top level of the conformance test source tree in the same way as the distributed tarball, with all generated files such as objects and libraries removed, and with a README-<company name> at the top summarizing the changed files. The annotations should make it clear what changes have been made and for what reason.
- The result log, README-<company name>, and statement of conformance must each be plain text files readable in a simple text editor.

G6. Passing Criteria

- Conformance can be claimed for one or more profiles of OpenSL ES (“Phone”, Music” and “Game”). A conformant implementation must unconditionally pass all tests for a profile in order to claim conformance with that profile.
- For the claimed profile(s):
  - All mandated minimum requirements (use case) tests must pass, as defined in Section 4.7 of the OpenSL ES specification.
  - All mandated object tests and their corresponding mandated interface tests must pass.
- For all optional functionality implemented, the optional objects and corresponding interfaces must also pass the respective conformance tests.

G7. Review Period

30 Days

G8. Conformant Product Criteria

Conformant Products must be similar to the Conformant Implementation in the following ways:

- the identical Specification implementation, i.e., identical binaries and/or accelerator data path to the component output, or new versions of the binaries and/or accelerator data path that do not cause any previously passing test to fail.
- the same major version of the same OS that uses substantially similar media processing functionality, or minor version updates to the OS that do not cause any previously passing test to fail.

Conformance tests would have to be re-run (but a re-submission of the results is not required) under the following circumstances:

- If a previously conformant OpenSL ES implementation is using an underlying conformant implementation of another Khronos API (e.g., OpenMAX IL) and there are changes made to that underlying implementation (that are subsequently validated by a re-run of that API's own conformance tests), then the OpenSL ES conformance tests should be re-run as a sanity check to ensure that changes to the underlying implementation have not caused any of the OpenSL ES conformance tests to fail.
- If a previously conformant OpenSL ES implementation is interacting with a proprietary implementation of another software module (e.g. DRM) and there are no interfaces into this module from OpenSL ES, then any changes to this software module (e.g. one DRM agent replaced by another one of similar functionality) would necessitate a re-run of the OpenSL ES conformance tests, as a sanity check, to ensure that the changes to this module do not cause any conformance test failures.

Conformance tests would have to re-run and results re-submitted under the following circumstances:

- If a previously conformant OpenSL ES implementation is interacting with a proprietary implementation of another software module (e.g., 3D Audio, purchasing) and there are interfaces into this software module from OpenSL ES, then any changes to this software module shall necessitate a re-run **and** re-submission of the OpenSL ES conformance tests to ensure that the changes to this module do not cause any conformance test failures.
- If a previously conformant OpenSL ES implementation is extended with additional functionality, such as extending support for an additional profile, it shall necessitate a re-run **and** re-submission of the OpenSL ES conformance tests covering the previous and new functionality to ensure that the added functionality does not cause any previously passed conformance test to fail, and that the new functionality is conformant.

G9. Marks and Usage Guidelines

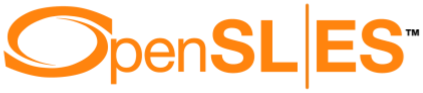
The following usage guidelines must be followed for any use of the Marks below:

- "™" must be used as shown with the first use of the written Mark in a document.

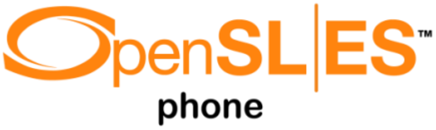
- the following text must be included in each document that uses the Marks: "OpenSL ES and the OpenSL ES logo are trademarks of the Khronos Group Inc."

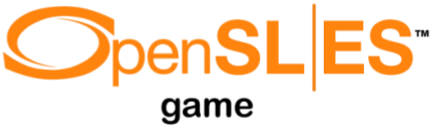
(i) OpenSL ES™

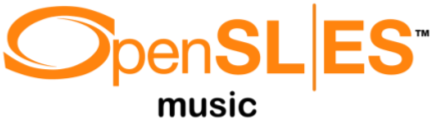
(ii) OpenSL ES Logo:

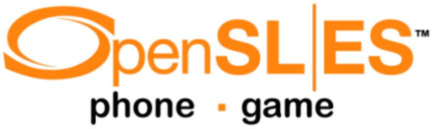


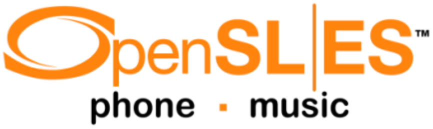
The above logo is for use in documents and presentations referring to the OpenSL ES API in general. When discussing a specific OpenSL ES implementation, one of the following profile combination specific logos must be used. The logo used must reflect the profiles claimed at the time of conformance:

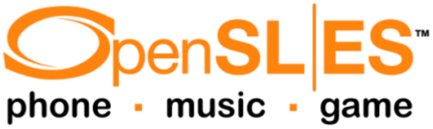












Attachment H
OpenMAX AL Conformance Process Details

H1. Change History and Version

- **June 2009** – OpenMAX AL 1.0 Adopters Program released
- **Sept 2009 –** Minor typos fixed. All tracked changes accepted. Logos added.

H2. Paid Specification Versions Covered

OpenMAX AL 1.0 including minor release updates (i.e., all OpenMAX AL 1.x versions are covered).

H3. Conformance Fee

	Khronos Member	**Non-member**
OpenMAX AL 1.0 and minor specification updates	$10K	$15K

H4. Tests Source Code Packaging

The source code for the Tests is packaged in a gnu-zipped tar file. It is named using the following format: conformes-<OpenMAX-AL version>r<revision number>-<year><month><day>.tgz. The revision number is incremented for every bug fix release of the Tests targeted at a specific version of the Specification. When the tar file is made, a tag is added to the conformance test SVN repository so that Adopters can sync the SVN tree against a particular bugfix release. The SVN tags use the following format: OMXAL-<major version>-<minor version>-rev<revision number>.

H5. Submission Package

A Submission must contain the information defined in the Submission section of the process document PLUS all the following Specification specific information:

- Identification of the Implementation including: the CPU running the Specification, the OS and the Specification pipeline – including version numbers.
- The result log (output) for the executed tests, run in the order and with the parameters specified in the "EXECUTION" section of the "README" file located at the top level of the conformance test source tree.
- The complete source of the executed tests together with an annotated diff file containing any source changes packaged as a ZIP archive file (.zip) or gnu zip compressed tarball (.tar.gz) rooted at the top level of the conformance test source tree in the same way as the distributed tarball, with all generated files such as objects and libraries removed, and with a README-<company name> at the top summarizing the changed files. The annotations should make it clear what changes have been made and for what reason.
- The result log, README-<company name>, and statement of conformance must each be plain text files readable in a simple text editor.

H6. Passing Criteria

- Conformance can be claimed for one or more profiles of OpenMAX AL (“Media Player” and “Media Player/Recorder”). A conformant implementation must unconditionally pass all tests for a profile in order to claim conformance with that profile.
- In addition to a profile or profiles, conformance can be claimed for the “+ MIDI” optional extension. A conformant implementation must unconditionally pass all tests for this extension to claim conformance with that extension.
- For claimed profile(s) all mandated object tests and their corresponding mandated interface tests must pass.
- For all optional functionality implemented, the optional objects and corresponding interfaces must also pass the respective conformance tests.

H7. Review Period

30 Days

H8. Conformant Product Criteria

Conformant Products must be similar to the Conformant Implementation in the following ways:

- the identical Specification implementation, i.e., identical binaries and/or accelerator data path to the component output, or new versions of the binaries and/or accelerator data path that do not cause any previously passing test to fail.
- the same major version of the same OS that uses substantially similar media processing functionality, or minor version updates to the OS that do not cause any previously passing test to fail.

Conformance tests would have to be re-run (but a re-submission of the results is not required) under the following circumstances:

- If a previously conformant OpenMAX AL implementation is using an underlying conformant implementation of another Khronos API (e.g., OpenMAX IL) and there are changes made to that underlying implementation (that are subsequently validated by a re-run of that API's own conformance tests), then the OpenMAX AL conformance tests should be re-run as a sanity check to ensure that changes to the underlying implementation have not caused any of the OpenMAX AL conformance tests to fail.
- If a previously conformant OpenMAX AL implementation is interacting with a proprietary implementation of another software module (e.g. DRM) and there are no interfaces into this module from OpenMAX AL, then any changes to this software module (e.g. one DRM agent replaced by another one of similar functionality) would necessitate a re-run of the OpenMAX AL conformance tests, as a sanity check, to ensure that the changes to this module do not cause any conformance test failures.

Conformance tests would have to be re-run and results re-submitted under the following circumstances:

- If a previously conformant OpenMAX AL implementation is interacting with a proprietary implementation of another software module (e.g., purchasing) and there are interfaces into this software module from OpenMAX AL, then any changes to this software module shall necessitate a re-run **and** re-submission of the OpenMAX AL conformance tests to ensure that the changes to this module do not cause any conformance test failures.
- If a previously conformant OpenMAX AL implementation is extended with additional functionality, such as extending support for an additional profile, it shall necessitate a re-run **and** re-submission of the OpenMAX AL conformance tests covering the previous and new functionality to ensure that the added functionality does not cause any previously passed conformance test to fail, and that the new functionality is conformant.

H9. Marks and Usage Guidelines

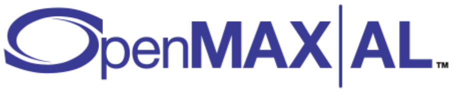
The following usage guidelines must be followed for any use of the Marks below:

- "™" must be used as shown with the first use of the written Mark in a document.

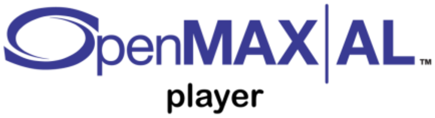
- the following text must be included in each document that uses the Marks: "OpenMAX AL and the OpenMAX AL logo are trademarks of the Khronos Group Inc."

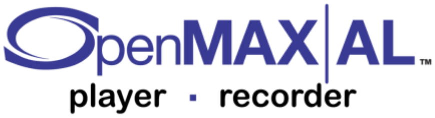
(i) OpenMAX AL™

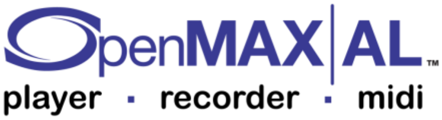
(ii) OpenMAX AL Logo:

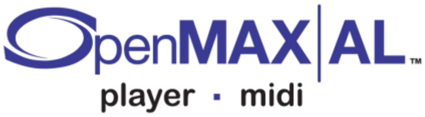


The above logo is for use in documents and presentations referring to the OpenMAX AL API in general. When discussing a specific OpenMAX AL implementation, one of the following profile combination specific logos must be used. The logo used must reflect the profiles claimed at the time of conformance:









Attachment I
COLLADA Conformance Process Details

I1. Change History and Version

- **Apr 2007** – first version attached to generic process document
- **Jun 2007** – second version with badge logos and implementation description
- **Sep 2009** – third version updated for first public release
- **Oct 2009** – fourth version with changes from Phoenix F2F
- **Dec 2009** – fifth version with updated badge logos and package descriptions
- **Oct 2010** – sixth version with updated submission package to point to external document. Updated conformance fees to match web site

I2. Paid Specification Versions Covered

This attachment covers the COLLADA specification version 1.4 including minor specification updates (e.g., COLLADA 1.4.0, and 1.4.1 are covered, etc.).

I3. Conformance Fees

The Conformance Fee and Upgrade Conformance Fee for each version of the Specification are listed below:

	Academic Member or Academic Adopter	**Khronos Member Fee**	**Non-member Fee**
COLLADA 1.4 and minor specification updates	$1K	$6k (Adopter’s gross revenue > $50M) $3k (Adopter’s gross revenue < $50M)	$11k (Adopter’s gross revenue > $50M) $8k (Adopter’s gross revenue < $50M)

I4. Tests Source Code Packaging

The source code for the Tests is packaged in a zip file. It is named using the following format:

COLLADA-<COLLADA version>-CTS<revision number>-<year><month><day>.zip

The revision number is incremented for every bug fix release of the Tests targeted at a specific version of the Specification. The date is also updated. For example:

COLLADA-1.4-CTS-0.9.0-20091210.zip

COLLADA-1.4-CTS-1.0.0-20100201.zip

COLLADA-1.4-CTS-1.0.1-20100207.zip

When the zip file is made, a tag is added to the conformance test SVN repository so that Adopters can sync the SVN tree against a particular bug fix release. The SVN tags use the following format:

COLLADA-<COLLADA version>-CTS<revision number>-<year><month><day>

For example:

COLLADA-1.4-CTS-0.9.0-20091210

COLLADA-1.4-CTS-1.0.0-20100201

COLLADA-1.4-CTS-1.0.1-20100207

I5. Submission Package

A Submission must contain the information defined in the Submission section of the Process Document PLUS all of the following Implementation specific information defined in the SubmittingResults.doc file in the Documentation folder. Upload the generated results.zip to the Submission Repository.

I6. Passing Criteria

A Conformant Implementation must unconditionally pass the “BASELINE” badge tests at minimum. Other tests are run and result in additional badge levels if fully passed.

An Implementation of COLLADA is one of the following:

- A standalone application that can import, export, and render COLLADA documents while running in a specific environment. The environment is specified by the following components: the OS (Linux, Mac OS X, Windows, …), COLLADA libraries (COLLADA DOM, Crosswalk SDK, FCollada, OpenCOLLADA, …) and supporting libraries (Java, libc, libxml2, Cocoa, MFC, MSXML, …) – including version numbers for all components.
- An application plugin or codec that can import, export, and render COLLADA documents while running in a specific environment. The environment is specified by the following components: the OS (Linux, Mac OS X, Windows, …), the application plugin SDK (Max, Maya, Softimage, …), COLLADA libraries (COLLADA DOM, Crosswalk SDK, FCollada, OpenCOLLADA), and supporting libraries (Java, libc, libxml2, Cocoa, MFC, MSXML, …) – including version numbers for all components.

A COLLADA library is one that parses and semantically understands COLLADA content.

I7. Review Period

30 Days.

I8. Conformant Product Criteria

Conformant Products must be similar to the Conformant Implementation in the following ways:

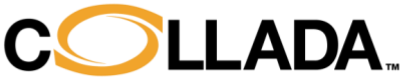
- The identical Implementation or the same major versions of the Implementation (as defined in Section H6. Passing Criteria) that contains only bug fixes and new features that do not cause any previously passing test to fail.
- The identical set or COLLADA specific subset of the Implementation that may be combined with code not used by the Implementation that does not cause any previously passing test to fail.

I9. Marks and Usage Guidelines

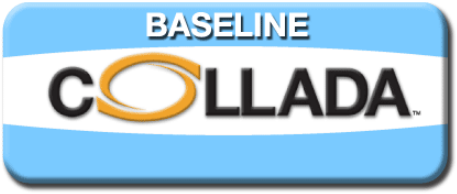
The following usage guidelines must be followed for any use of the Marks below:
- "™" must be used as shown with the first use of the written Mark in a document
- the following text must be included in each document that uses the Marks: "COLLADA, the COLLADA logo and COLLADA badge logos are trademarks of the Khronos Group Inc."

(v) COLLADA™

(vi) COLLADA™ Logo:

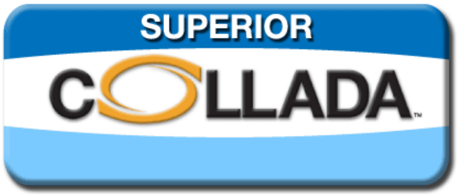


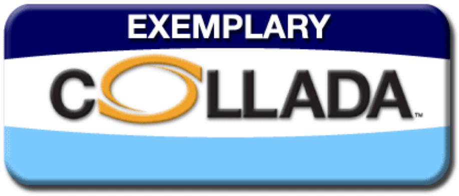
(vii) COLLADA™ BASELINE Logo:



(viii) COLLADA™ SUPERIOR Logo:

(ix) COLLADA™ EXEMPLARY Logo:





Attachment J
OpenWF Conformance Process Details

J1. Change History and Version

- **November 2008** – first version attached to generic process document
- **3 August 2009 –** Added section on changes to test thresholds
- **25 September 2010** – Added conformance fees
- **July 2016** – clarified that member discount applies to all member classes

J2. Paid Specification Versions Covered

OpenWF Display 1.0 including minor release updates.

OpenWF Composition 1.0 including minor release updates.

J3. Conformance Fees

Conformance Fees for each version of the Specification are given in the table below. Note that Fees for a given version automatically include Fees for all previous versions. New Adopters pay the Nonmember Base Fee for the version they are adopting.

	Base conformance fee (Nonmember)	**Base conformance fee (Member)**
OpenWF 1.0	$15K	$10K

J4. Test Source Code Packaging

J4.1 OpenWF Composition

The source code for the OpenWF Conformance Tests is packaged in a gnu-zipped tar file.

The file name has the format:
WFC-CTS-<X.Y>-<Z>.<W>-<year><month><day>.tgz.

Here <X.Y> is the version of the OpenWF Composition Specification to which the test applies, <Z> is the major revision number of the conformance test, and <W> is the minor revision number of the test. Changes in minor revision number reflect changes intended to correct bugs or improve portability and maintainability of the tests.

Changes in major revision number reflect changes that significantly expand test coverage and/or impose stricter Passing Criteria. For example, a test labeled WFC-CTS-1.0-1.0-20090201.tgz applies to OpenWF Composition 1.0, and is version 1.0 of the conformance test, released on February 1st, 2009.

When a new test release is created, a tag is added to the conformance test Subversion (SVN) repository so that Adopters who wish to can sync the SVN tree against a particular bugfix release. The SVN tag for a release has the same name as the .tgz file, without the .tgz extension.

As specified in Section 5 of the Conformance Process Document, Adopters may make Submissions using any version of the Tests and Process but are encouraged to use the most recent version that their implementations are able to pass.

All versions of the OpenWF 1.X -source code tree contain a top-level file referred to in this document as “the README file”. Those files define Submission Package format and Passing Criteria and provide detailed instructions for porting the tests and running them to generate conformance results.

J4.2 OpenWF Display

The source code for the OpenWF Conformance Tests is packaged in a gnu-zipped tar file.

The file name has the format:

WFD-CTS-<X.Y>-<Z>.<W>-<year><month><day>.tgz.

Here <X.Y> is the version of the OpenWF Display Specification to which the test applies, <Z> is the major revision number of the conformance test, and <W> is the minor revision number of the test. Changes in minor revision number reflect changes intended to correct bugs or improve portability and maintainability of the tests.

Changes in major revision number reflect changes that significantly expand test coverage and/or impose stricter Passing Criteria. For example, a test labeled WFD-CTS-1.0-1.0-20090201.tgz applies to OpenWF Display 1.0, and is version 1.0 of the conformance test, released on February 1st, 2009.

When a new test release is created, a tag is added to the conformance test Subversion (SVN) repository so that Adopters who wish to can sync the SVN tree against a particular bugfix release. The SVN tag for a release has the same name as the .tgz file, without the .tgz extension.

As specified in Section 5 of the Conformance Process Document, Adopters may make

Submissions using any version of the Tests and Process but are encouraged to use the most recent version that their implementations are able to pass.

All versions of the OpenWF 1.X -source code tree contain a top-level file referred to in this document as “the README file”. Those files define Submission Package format and Passing Criteria and provide detailed instructions for porting the tests and running them to generate conformance results.

J5. Submission Package

The format and contents of a valid Submission Package are defined in the SUBMISSION PACKAGE section of the README file included in the conformance test suite package.

J6. Passing Criteria

A Conformant Implementation must satisfy the requirements specified in the EXECUTION and PASSING CRITERIA sections of the README file included in the conformance test suite package.

J7. Review Period

30 Days

J8. Conformant Product Criteria

Conformance may be claimed for any component-level products necessary for a Conformant Implementation.

Implementations claimed as Conformant Products must be similar to the Conformant Implementation in the following ways:

- the identical rendering pipeline – i.e., identical binaries and/or accelerator data path to the display, or if a JIT compiler is used to generate binaries, then the use of the identical JIT compiler binary, or new versions of the binaries and/or accelerator data path or JIT compiler binary that do not cause any previously passing test to fail.
- the same major version of the same OS that uses substantially similar display functionality or minor version updates to the OS that do not cause any previously passing test to fail.
- the identical set, or a subset, of supported configurations. Different display resolutions are permitted.

J9. Marks and Usage Guidelines

The following usage guidelines must be followed for any use of the Marks below:

- “™” must be used as shown with the first use of the written Mark in a document

- the following text must be included in each document that uses the Marks: “OpenWF is a registered trademark and the OpenWF Display and OpenWF Composition logos are trademarks of Khronos.”

(i) OpenWF™ Composition

(ii) OpenWF™ Display

(iii) OpenWF Composition logo:

OpenWF™
COMPOSITION

(iii) OpenWF Display logo:

OpenWF™
DISPLAY

J10. Extending conformance tests to support new color formats

If the implementation supports colors formats not covered by conformance test suite, the adopter must extend the conformance test suite to support the new color formats and apply for the appropriate waivers.

The conformance submission must include all modifications to conformance tests required to support the new formats. These changes include:

- any modifications to test case and test client source code
- any modifications to conversion routines used by tests
- any modification to off-device test configurations and source code
- new off-screen test reference images (not normally required)

New reference images should only be required in exceptional circumstances as most new formats should be comparable to existing reference images using the appropriate tolerance.

It is important for new formats to accurately report their precision (number of colors per channel) via the on-device framework so that the tolerances used when comparing to reference images remain strict.

J11. Fixing bugs in conformance tests

Other than changes needed for porting, the only changes that are permitted are changes to fix bugs in the conformance test. A bug in the conformance test is a behavior which causes clearly incorrect execution (e.g., hanging, crashing, or memory corruption), OR a passing criterion which exceeds what is required by the relevant OpenWF Specification. Changes required to address either of these issues typically require waivers.

Threshold Changes

Changes to thresholds used to compare test results with reference images must be approved by the working group. In general, any weakening of thresholds should be limited to specific sub-tests and specific format categories, rather than weakening the test's default threshold.

Requests to weaken thresholds must be accompanied by images the requestor considers to be "good" that cause the tests to fail when using the old threshold values. Requests to strengthen thresholds must be accompanied by images the requestor considers to be "bad" that cause the tests to succeed when using the old threshold values. The resolution of these images must match that of the default reference images provided with the conformance tests, so they are directly comparable.

J12. Waivers

The procedure to for requesting a waiver is to report the issue by filing a bug report in Bugzilla (https://cvs.khronos.org/bugzilla/). When you create your submission package, include the waivers as described in the Adopters agreement. Including as much information as possible (including attachments of suggested file changes) will ensure the issue can be progressed as speedily as possible.

Attachment K
OpenVX Conformance Process Details

K1. Change History and Version

- **October 2014** – first release
- **July 2016** – added OpenVX 1.1
- **November 2017 –** added OpenVX 1.2, removed OpenVX 1.0
- **August 2019** – Added OpenVX 1.3

K2. Paid Specification Versions Covered

The fees below include all minor specification updates. Paid fees do not cover specifications with unreleased tests at the time of payment, e.g.: if fees were paid for OpenVX 1.1 or 1.2 before OpenVX 1.3 tests were released, OpenVX 1.3 fees from the table below must be paid to enable OpenVX 1.3 submissions.

K3. Conformance Fees

The Conformance Fees for each version of the Specification is below:

	Khronos Member	**Non-Member**
OpenVX 1.1.X, 1.2.X, and 1.3.X	$20K	$30K

The Adopter's program for OpenVX 1.0.X is closed. submissions for 1.0.X conformance will no longer be accepted. New submissions must be for 1.1.X, 1.2.X, or 1.3.X.

K4. Tests Source Code Packaging

The source code for the conformance tests is packaged in a bzip2 tar file. It is named using the following format:

OpenVX-CTS-<API version>-<year><month><day>.tar.bz2

The date field is updated with each bug fix release of the conformance tests.

When the tar file is made, a tag is created to the conformance test git repository so that Adopters can check out the git tree against a particular bug fix release. The git tags use the following convention:

- In case of a full OpenVX release (headers, specification, conformance tests):

  RELEASE_<API version>_<month>_<day>_<year>

- In case of an incremental release of the conformance tests:

  CTS_<API version>_<month>_<day>_<year>

Adopters are encouraged to use the latest revision of the CTS, but may make submissions using an older version, unless the older version has been withdrawn. Khronos reserves the right to withdraw older versions and to reject new submissions using those versions, provided that a) a newer version of the test has been available for at least 180 calendar days, and b) notice of intent to withdraw the older version was published at least 180 days before the effective date. Notice of intent to withdraw any test version, or of any change to the conformance test process, will be distributed by email to the OpenVX Adopters email list. A list of currently accepted test versions, together with dated notice of any intent to withdraw any of those versions, is available at https://www.khronos.org/registry/OpenVX.

K5. Submission Package

A Submission must contain the information defined in the Submission section (section 7) of this process document PLUS all of the following Specification specific information:

- The target System (i.e., SoC, Host processor with Accelerator) and OS (including version number) used to generate the Submission.
- The complete result log (output) for the executed tests, run in the order and with the parameters specified in the "EXECUTION" section of the "README" file located at the top level of the conformance test source tree.
- The complete source of the executed tests together with an annotated diff file containing any source changes Adopters Packaged as a ZIP archive file (.zip) or gnu zip compressed tarball (.tar.gz) rooted at the top level of the conformance test source tree in the same way as the distributed tarball, with all generated files such as objects and libraries removed, and with a README-<company name> at the top summarizing the changed files. The annotations should make it clear what changes have been made and for what reason.
- The result log, README-<company name>, and statement of conformance must each be plain text files readable in a simple text editor.
- For OpenVX 1.3 submissions, an indication of the conformance feature set(s) being tested, which must reflect the selections made when building the CTS

K6. Passing Criteria

Conformant implementation must pass all test cases for the selected conformance feature set(s)

K7. Review Period

30 Days

K8. Conformant Product Criteria

Conformant Products must be similar to the Conformant Implementation in the following ways:

- the identical Specification implementation, i.e., identical binaries and/or accelerator data path to the component output, or new versions of the binaries and/or accelerator data path that do not cause any previously passing test to fail.
- the same major version of the same OS that uses substantially similar media processing functionality, or minor version updates to the OS that do not cause any previously passing test to fail.

K9. Marks and Usage Guidelines

The following usage guidelines must be followed for any use of the Marks below:

- “™” must be used as shown with the first use of the written Mark in a document.

- the following text must be included in each document that uses the Marks: “OpenVX and the OpenVX logo are trademarks of the Khronos Group Inc.”

(i) OpenVX™

(ii) OpenVX Logo:

Attachment L
WebGL Conformance Process Details

L1. Change History and Version

- **November 2015** – first release
- **January 2017** – added WebGL 2.0

L2. Paid Specification Versions Covered

WebGL 1.0 and 2.0

L3. Conformance Fees

The Conformance Fee for each version of the Specification is below:

	Member Base Fee	Non-Member Base Fee	Upgrade Fee
WebGL 1.0	$0	$0	NA
WebGL 2.0	$0	$0	NA

L4. Tests Source Code Packaging

The test source code package is contained in the GitHub repo:
https://github.com/KhronosGroup/WebGL

At:
https://github.com/KhronosGroup/WebGL/tree/master/conformance-suites

Submissions should use the latest uploaded version.

L5. Submission Package

Instructions for creating and uploading a Submission Package are located at:
https://github.com/KhronosGroup/WebGL/blob/master/sdk/tests/README.md

L6. Passing Criteria

A conformant implementation must pass all test cases

L7. Review Period

30 Days

L8. Conformant Product Criteria

(A) Conformance Rules for a Web Browser Implementer

1. Conformance on a particular operating system

On a given OS, a WebGL implementation will be considered to conform to a particular version of the conformance suite if the suite passes with no test failures on at least two GPUs, each from a different vendor. If the OS only supports a GPU from one vendor, the two-GPU requirement is dropped.

2. Conformance across multiple operating systems

A WebGL implementation will be considered to conform to a particular version of the conformance suite if it passes rule (1) on all of the OSs on which the WebGL implementation is intended to be supported.

3. Conformance as the web browser is upgraded

WebGL conformance results submitted for an earlier version of the browser carry forward to later versions of the browser that do not cause any previously passing test to fail.

4. Conformance as the operating system is upgraded

If a new version is released of one of the OSs on which a WebGL implementation is intended to run, then WebGL conformance results submitted for earlier versions of that OS carry forward. Future conformance results must be submitted against the new version of the OS. If it is anticipated that the older OS version will be supported for some time, then future conformance results must be submitted separately for both the old and new versions of the OS.

(B) Conformance Rules for a GPU Vendor

A GPU vendor submitting conformance results for a WebGL implementation typically does so because the device containing the GPU includes a built-in web browser. In this case the following rules apply:

1. Conformance results must be submitted for each GPU and operating system combination to be certified. It is not required to submit results for different devices containing the same GPU and running the same operating system that do not cause any previously passing test to fail.

2. Conformance results carry forward for a given GPU as the operating system and graphics driver are upgraded but do not cause any previously passing test to fail.

L9. Marks and Usage Guidelines

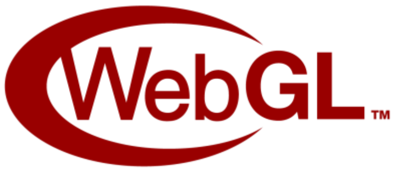
The following usage guidelines must be followed for any use of the Marks below:

- "™" must be used as shown with the first use of the written Mark in a document.

- the following text must be included in each document that uses the Marks: "WebGL and the WebGL logo are trademarks of the Khronos Group Inc."

(i) WebGL™

(ii) WebGL Logo:



Attachment M
Vulkan Conformance Process Details

M1. Change History and Version

- **February 16, 2016** – first version
- **July 2016** – updates and corrections
- **March 2018** – added Vulkan 1.1
- **September 2019** – added Vulkan 1.2
- **December 2021** - changed pricing model to subscription

M2. Paid Specification Versions Covered

(1) For Adopters who adopted Vulkan subject to the one-off fee model per V23 or prior of the Khronos Adopters Agreement ("Prior Adopters"): the version(s) of Vulkan specifically selected and paid for and associated minor updates.

(2) For Adopters who adopt Vulkan subject to the subscription fee model per V24 or later of the Khronos Adopters Agreement: all published versions (including extensions and major or minor revisions)

M3. Conformance Fees

For Prior Adopters: Adopters may make Submissions for any version of Vulkan up to and including the Paid Specification Version specified in M2.(1) above for which they had already paid the applicable Adopters fee, without paying additional fees. There is no Expiration Date that applies to such Adopters.

For Adopters who adopt Vulkan per V24 or later of the Adopters Agreement: Vulkan Conformance Fees are based on a subscription model rather than a one-time per-version Fee. These Conformance Fees for subscriptions include Subscription Fees and Discounted Fees, as follows:

- The initial Subscription Fee is due when an Adopter's Agreement is executed for Vulkan. There is a one-time discount for the initial Subscription Fee for Prior Adopters who sign a new Adoption Agreement with the new subscription model by or before 12/31/2022. Adopter's initial subscription term, and the applicable Expiration Date, will be set at one calendar year after Khronos receives the initial Subscription Fee.
- To keep the Adopter's subscription current, Adopter must pay the Renewal Fee by or before the applicable Expiration Date, upon which payment the Adopter's subscription term will be extended by one calendar year.
- If an Adopter allows its subscription to lapse, its ability to make Submissions will be suspended (except to the extent permitted as a Prior Adopter as described below), unless and until it pays the Subscription Fee. Upon Khronos' receipt of payment of a Subscription Fee from a lapsed Adopter that Adopter's subscription term will be set at one calendar year from the payment date.
- The date upon which the subscription term expires is the Expiration Date.
- Khronos may allow a grace period for an Adopter's eligibility for Discounted Fees, at its discretion.

Current Subscription Fees and Renewal Fees are as follows. Adopters who are current members of Khronos at the time of payment are eligible for the Member Fee rates. All other Adopters must pay the Non-Member Fee rates:

	Member	**Non-Member**
Subscription Fee	$70K ($45K for Prior Adopters Available until 12/31/2022)	$90K ($55K for Prior Adopters Available until 12/31/2022)
Renewal Fee	$45K	$55K

Khronos may change this fee schedule at any time at its sole discretion, but changes shall not take effect until one calendar year after notice is given to all applicable Adopters.

M4. Test Source Code Packaging

The Vulkan Conformance Test Suite (CTS) is revised periodically to improve coverage and to fix bugs. Official releases are identified by version strings of the form *vulkan-cts-X.Y.Z.W*, where *X.Y* is the latest version of the Vulkan API that the test applies to, and *Z.W* is the revision number of the test.

Adopters are encouraged to use the latest revision of the CTS available. All versions of the CTS apply to all previous versions of the API, so *vulkan-cts-1.2.0.1*, if it exists, is a more recent CTS for Vulkan 1.1 implementations than any version named *vulkan-cts-1.1.Z.W*. Adopters may make submissions using an older version, unless the older version has been withdrawn. Khronos reserves the right to withdraw older versions and to reject new submissions using those versions, provided that a) a newer version of the test has been available for at least 180 calendar days, and b) notice of intent to withdraw the older version was published at least 180 days before the effective date.

Notice of intent to withdraw any test version, or of any change to the conformance test process, will be distributed by email to the Vulkan Adopters email list. A list of currently accepted test versions, together with dated notice of any intent to withdraw any of those versions, is available at https://github.com/KhronosGroup/VK-GL-CTS/wiki.

The source code for the Conformance Tests is maintained as an open-source project at https://github.com/KhronosGroup/VK-GL-CTS. Official releases are represented by tagged commits to that repository, using the version string as the tag. To obtain a copy of the source code for any released version of the test, install `git` and execute the command:

```
$ git clone https://github.com/KhronosGroup/VK-GL-CTS.git -b <version>
```

where `<version>` is the version string associated with the desired release. Versions of the source that do not carry a release tag may not be used to make conformance submissions.

M5. README file

All versions of the Vulkan conformance test contain a file named (from the root directory) *external/vulkancts/README.md*, referred to in this document as “the README file”. The README file defines Submission Package format and Passing Criteria for the version in question and provides instructions for porting the tests and running them to generate conformance results.

M6. Submission Package

The format and contents of a valid Submission Package are defined in the ***Conformance Submission Package Requirements*** section of the README file.

M7. Passing Criteria

Tests must be run as described in the ***Running CTS*** section of the README file and the results must satisfy the requirements given in the ***Conformance Criteria*** section of that file.

M8. Review Period

The review period for ordinary Submissions is 30 calendar days. The working group may approve shorter review periods in special situations, such as prior to a new specification release.

The review period for Submission Updates and Reference Submissions is 7 calendar days.

The Vulkan Working Group may, at its sole discretion, vote to waive the balance of the Review Period and declare a Submission conformant at any point during the Review Period.

If the Adopter’s Expiration Date passes during the Review Period for a submission, it has no effect on the Review Period or any other aspect of the review process. The Adopter may still make Submission Alterations, Submission Corrections, and Submission Deletions related to that Submission, but may no longer make Submission Updates or Reference Submissions based on it.

M9. Conformant Product Criteria

Conformant Products must be similar to the Conformant Implementation in the following ways:

(i) For non-layered implementations: the same hardware rendering pipeline, i.e., the same accelerator chipset or SoC, or a variant chipset or SoC using the same accelerator core, or minor revisions to the above that do not cause any previously passing test to fail; the same control software, i.e., drivers built from the same driver source, or minor revisions to the above that do not cause any previously passing test to fail; the same set of exposed physical devices. Corresponding devices must be identical to the reference in terms of physical device features, device limits, queues and queue properties, and supported extensions. Implementations differing only in number of cores are still considered to make use of the same rendering pipeline.

(ii) For layered implementations over a certified DX12 or production Metal API: the same implementation software (i.e. built from the same source) and the same version of the underlying API including minor revisions to either that do not cause any previously passing test to fail, and for each OS, there must be Successful Submissions using at least two (if available) independent implementations of the underlying API from different vendors (if available), at least one of which must be hardware accelerated.

(iii) The same OS, or a derivative version of the OS, or minor revisions to the above that do not cause any previously passing test to fail.

(iv) The same functionality, i.e., all tests that pass on the Conformant Implementation must also pass on Conformant Products.

Conformance may be claimed for any component-level products necessary for a Conformant Implementation.

M10. Mark and Logo Usage Guidelines

The following usage guidelines must be followed for any use of the Marks identified below:

- The Vulkan Mark and Logo may be used only with respect to Conformant Products that have passed the Vulkan Conformance Process, and that support the features of any extensions or profiles that the Conformant Products are claimed to support.
- "®" must be used as shown below with the first use of the written Mark in a document.
- Each document that uses the Marks must include the following text: "Vulkan and the Vulkan logo are registered trademarks of the Khronos Group Inc."

The Marks are:

- Vulkan®
- The Vulkan Logo:

Attachment N
SYCL Conformance Process Details

N1. Change History and Version

- **June 2018** – SYCL 1.2.1 Adopters Program added
- **November 2023** – SYCL 2020 Fees and layered implementations added

N2. Paid Specification Versions Covered

This attachment covers the conformance process for all versions of SYCL specification from SYCL 1.2.1 up to SYCL 2020, including minor specification updates prior to the next major specification release. This attachment does not cover the conformance process for SYCL 1.2 or versions of the conformance tests released before 5th July 2018. Paid fees do not cover specifications with unreleased tests at the time of payment, e.g.: If fees were paid for SYCL 1.2.1 before SYCL 2020 tests were released, SYCL 2020 fees from the table below must be paid to enable SYCL 2020 submissions.

N3. Conformance Fee Schedule

	Academic Member or Academic Adopter Base Fee	**Khronos Member Base Fee**	**Non-Member Base Fee**
SYCL 1.2.1 **and minor specification updates**	$0	$15,000	$20,000
SYCL 2020 **and minor specification updates**	$0	$30,000	$40,000

N4. Test Source Code Packaging

The source code for the conformance tests shall be taken from the public SYCL CTS GitHub repository: https://github.com/KhronosGroup/SYCL-CTS. A conformance submission can be made using any commit from this repository within the SYCL-1.2.1/master or SYCL-2020 branches, depending on which version of the SYCL specification the submission is for, where the commit date is after 28th April 2023 and within 3 months of the submission date. If any patches to the chosen commit are required in order to fix a bug, these patches must be cherry-picked from either subsequent commits within the same branch or from commits in an open pull request, without merge conflicts. Any required patches must be documented in the submission as a list of commit SHAs.

Note that if commits are cherry-picked from an open pull request, these commits are still to be approved by the working group and therefore could cause a delay or even rejection of a submission, so it's encouraged for these pull requests to be merged into the target branch before submission.

N5. Submission Package

A Submission must contain the following information contained in the top-level directory of a zip file:

- Submission information in a file called "submission_details.txt" in the format defined in the SYCL Implementation Details Template document "submission_details_template.txt". The template can be found in the conformance package under the test_conformance directory;
- A file called "build_information.txt" that describing the build procedure (CMake, environment, toolchain, etc), the branch and commit SHA which was used and a list of commit SHAs for every source change

required to fix any bugs describing whether the change is from a subsequent commit of the same branch or from a pull request, the changes made and the reason.

- The submission_details.txt and build_information.txt files must each be plain text files readable in a simple text editor;
- All result logs for all executed tests. Conformance report in a file called “conformance_report.xml” generated by the “run_conformance_tests.py” script according to the instructions in “README.md”. This report will contain details of the host system, the target SYCL backend, platform and device, , the parameters used to run the tests and all test logs and results;
- The full test directory from the conformance package including any generated source files and header files.

A submission may skip certain tests in the case that they cannot be passed due to limitations in the host architecture or the target SYCL backend, platform and device. Justification must be provided as to why these tests were skipped and this will be reviewed and approved by the SYCL working group during the Review Period.

N6. Passing Criteria

A Successful Submission must contain a “conformance_report.xml” file generated by running “run_conformance_tests.py”, containing details of the host architecture and OS, the target SYCL backend, the target SYCL device and platform, a list of optional and KHR extensions features enabled, the parameters used to invoke the tests and the test logs from all categories run.

The conformance report must indicate that all mandatory categories were run and all categories which were run passed in full and did not contain any errors or failures targeting the specified SYCL device, platform and backend.

A Successful Submission must contain a description of SYCL platforms and devices which the Implementation supports. The description may specify all SYCL platforms and devices of the target SYCL backend, including future platforms and/or devices, or a subset of those, and must contain the target SYCL platform and device.

N7. Review Period

The default Review Period for Submissions, Submission Alterations and Submission Corrections is 30 calendar days. The default review period for Submission Updates and Reference Submissions is 7 calendar days.

The working group may approve shorter Review Periods for all Submissions, during a working group-agreed period, such as prior to a new specification release.

The working group may, at its sole discretion, vote to waive the balance of a Review Period to convert any type of Submission to a Successful Submission at any time.

N8. Conformant Product Criteria

Conformant Products may contain any number of SYCL backends, platforms and devices, however are only considered conformant for SYCL backends for which there is a Conformant Product targeting that backend and for SYCL platforms and devices which are described in a Successful Submission.

Conformant Products must be similar to a Conformant Implementation in the following ways:

- Each Conformant Product must have an identical or comparable specification implementation, i.e. identical specification binaries or a new version of the binaries that do not cause any previously passing tests to fail.
- Each Conformant Product must target the same major version of the same host OS or a different major version or different host OS that does not cause any previously passing tests to fail.
- Each Conformant Product must target the same host architecture or an update to the same host architecture that does not cause any previously passing tests to fail.
- Each Conformant Product must pass the tests for all SYCL backends, platforms and devices described in all Successful Submissions for the Conformant Implementation.

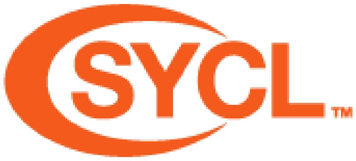
N9. Marks and Usage Guidelines

The following usage guidelines must be followed for any use of the Marks below:

- “TM” must be used as shown with the first use of the written Mark in a document
- the following text must be included in each document that uses the Marks: “SYCL and the SYCL logo are trademarks used by Khronos.”

(i) SYCL™

(ii) SYCL Logo:



Attachment O
OpenXR Conformance Process Details

O1. Change History and Version

- August 2019 – OpenXR 1.0 First Draft
- July 2020 – OpenXR 1.0 Public Release

O2. Paid Specification Versions Covered

OpenXR 1.0 and patch updates.

O3. Conformance Fees

Conformance Fees for OpenXR are given in the table below:

	Khronos Member	**Non-member**
OpenXR 1.0.X	$25K	$30K

O4. Test Source Code Packaging

The OpenXR Conformance Test Suite (CTS) is revised periodically to improve coverage and to fix bugs. Adopters are encouraged to use the latest revision of the CTS, but may make submissions using an older version, unless the older version has been withdrawn. Khronos reserves the right to withdraw older versions and to reject new submissions using those versions, provided that a) a newer version of the test has been available for at least 180 calendar days, and b) notice of intent to withdraw the older version was published at least 180 days before the effective date.

Notice of intent to withdraw any test version, or of any change to the conformance test process, will be distributed by email to the OpenXR Adopters email list. A list of currently accepted test versions, together with dated notice of any intent to withdraw any of those versions, is available at https://github.com/KhronosGroup/OpenXR-CTS/wiki.

O5. Test Source Code Packaging

The source code for the Conformance Tests is maintained as an open source project at https://github.com/KhronosGroup/OpenXR-CTS. Official releases are represented by tagged commits to that repository, using the version string as the tag. To obtain a copy of the source code for any released version of the test, install `git` and execute the command:

```
$ git clone https://github.com/KhronosGroup/OpenXR-CTS.git -b <version>
```

where `<version>` is the version string associated with the desired release. Versions of the source that do not carry a release tag may not be used to make conformance submissions.

O6. README file

All versions of the OpenXR conformance test contain a top-level file named README.md, referred to in this document as "the README file". The README file defines Submission Package format and Passing Criteria for the version in question and provides instructions for porting the tests and running them to generate conformance results.

O7. Submission Package

The format and contents of a valid Submission Package are defined in the ***Conformance Submission Package Requirements*** section of the README file.

O8. Passing Criteria

Tests must be run as described in the ***Running CTS*** section of the README file and the results must satisfy the requirements given in that file.

O9. Review Period

The review period for ordinary Submissions is 30 calendar days. The working group may approve shorter review periods in special situations, such as prior to a new specification release.

The review period for Submission Updates and Reference Submissions is 7 calendar days.

The OpenXR Working Group may, at its sole discretion, vote to waive the balance of the Review Period and declare a Submission conformant at any point during the Review Period.

O10. Conformant Product Criteria

Conformant Products must be similar to the Conformant Implementation in the following ways:

(i)The same runtime, i.e., runtimes built from the same source, or minor revisions to the above that do not cause any previously passing test to fail.

(ii) The same OS, or a derivative version of the OS, or minor revisions to the above that do not cause any previously passing test to fail.

(iii) The same functionality, i.e., all tests that pass on the Conformant Implementation must also pass on Conformant Products.

Conformance may be claimed for any component-level products necessary for a Conformant Implementation.

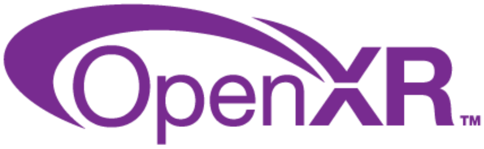
O11. Mark and Logo Usage Guidelines

The following usage guidelines must be followed for any use of the Marks identified below:

- The OpenXR Mark and Logo may be used only with respect to Conformant Products that have passed the OpenXR Conformance Process.
- “TM” must be used as shown below with the first use of the written Mark in a document.
- Each document that uses the Marks must include the following text: “OpenXR and the OpenXR logo are trademarks of the Khronos Group Inc.”

The Marks are:

- OpenXR™
- The OpenXR Logo:



Attachment P
Vulkan SC Conformance Process Details

P1. Change History and Version

- December 1, 2021 – first version

P2. Paid Specification Versions Covered

Vulkan SC 1.0, and minor updates (1.0.x).

P3. Conformance Fees

Conformance Fees for each version of the Vulkan SC Specification are given in the table below. As described in Section 4 above, Adopters may make Submissions for any version of Vulkan SC up to and including the Paid Specification Version without paying additional Fees. New Adopters who are Contributing or Promoting members of Khronos pay the Member Fee. All other new Adopters pay the Non-member Fee for the version they are adopting. There is no discount for Adopters who are upgrading from an older version of the API:

	Khronos Member	**Non-member**
Vulkan SC 1.0.X	$60K	$80K

P4. Test Source Code Packaging

The Vulkan SC Conformance Test Suite (CTS) is revised periodically to improve coverage and to fix bugs. Official releases are identified by version strings of the form vulkansc-cts-X.Y.Z.W, where X.Y is the latest version of the Vulkan SC API that the test applies to, and Z.W is the revision number of the test.

Adopters are encouraged to use the latest revision of the CTS available. All versions of the CTS apply to all previous versions of the API, so vulkansc-cts-1.1.0.0, if it exists, is a more recent CTS for Vulkan SC 1.0 implementations than any version named vulkansc-cts-1.0.Z.W. Adopters may make submissions using an older version, unless the older version has been withdrawn. Khronos reserves the right to withdraw older versions and to reject new submissions using those versions, provided that a) a newer version of the test has been available for at least 1 year, and b) notice of intent to withdraw the older version was published at least 1 year before the effective date.

Notice of intent to withdraw any test version, or of any change to the conformance test process, will be distributed by email to the Vulkan SC Adopters email list. A list of currently accepted test versions, together with dated notice of any intent to withdraw any of those versions, is available at https://github.com/KhronosGroup/VK-GL-CTS/wiki.

The source code for the Conformance Tests is maintained as an open-source project at https://github.com/KhronosGroup/VK-GL-CTS. Official releases are represented by tagged commits to that repository, using the version string as the tag. To obtain a copy of the source code for any released version of the test, install git and execute the command:

```
$ git clone https://github.com/KhronosGroup/VK-GL-CTS.git -b <version>
```

where `<version>` is the version string associated with the desired release. Versions of the source that do not carry a release tag may not be used to make conformance submissions.

While Vulkan SC CTS demonstrates conformance to the Vulkan SC API, it is not qualified for safety critical certification evidence purposes.

P5. README file

All versions of the Vulkan SC conformance test contain a file named (from the root directory) external/vulkancts/README.md, referred to in this document as “the README file”. The README file defines Submission Package format and Passing Criteria for the version in question and provides instructions for porting the tests and running them to generate conformance results.

P6. Submission Package

The format and contents of a valid Submission Package are defined in the **Conformance Submission Package Requirements** section of the README file.

P7. Passing Criteria

Tests must be run as described in the **Running CTS** section of the README file and the results must satisfy the requirements given in the **Conformance Criteria** section of that file.

P8. Review Period

The review period for ordinary Submissions is 30 calendar days. The working group may approve shorter review periods in special situations, such as prior to a new specification release.

The review period for Submission Updates and Reference Submissions is 7 calendar days.

The Vulkan SC Working Group may, at its sole discretion, vote to waive the balance of the Review Period and declare a Submission conformant at any point during the Review Period.

P9. Conformant Product Criteria

Terminology

- **Platform** – Refers to the CPU family and GPU family utilized to execute an implementation.
- **System** – Refers to the Platform and Operating System utilized to execute an implementation.
- **Reference System** – System utilized to execute the Conformance Test Suite on an implementation
- **Reference Implementation** – The Vulkan SC implementation that is used on the Reference System that the Conformance Test Suite is executed against
- **Target System** – Embedded ‘deployment’ system for an implementation (this may be production or representative hardware)
- **Target Implementation** – The Vulkan SC implementation that is used on the Target System that the device applications are executed against

Ideally, the Vulkan SC CTS is run directly on the Target Implementation using the Target System. However, in many scenarios the Target System for the implementation may be incapable of running the Vulkan SC CTS, due to limitations on compilers, operating systems, connectivity, storage, or other resources. In such situations conformance is instead permitted to be demonstrated by a Conformant Implementation running on a Reference System where:

1. The Reference Implementation and Target Implementation may differ in the following areas:

   i) Changes related to hardware differences between the Reference and Target Systems. For example, changes available memory on a platform, etc.

   ii) Changes related to Operating System differences between the Reference and Target Systems. For example, the Target System may utilize a safety critical OS (or no OS at all), while the Reference System may utilize Windows or Linux, etc.

   iii) Reduction in Vulkan SC API Support for the Target Implementation compared with the Reference Implementation. For example, removal of support for an extension or features or reduction of limits due to system constraints, etc.

   iv) Removal of API functionality for the Target Implementation compared with the Reference Implementation. For example, removal of Vulkan SC API functionality that is not required by the application(s) using the target implementation or to reduce safety certification burden, etc. The remaining Vulkan SC functionality in the Target Implementation used by the application must remain compatible with the Vulkan SC specification.

   v) In exceptional circumstances, additional Vulkan SC API Functionality in the Target Implementation not present in the Reference Implementation.

2. The CPU for both the Reference System and Target System are from the same family.

3. The GPU for both the Reference System and Target System are from the same family.

4. The Implementer has suitably tested their Implementation on both the Reference System and Target System, with the Implementer's test suite successfully executing on both systems. It is anticipated that safety critical Implementers will have substantial test suites to meet safety critical certifications.

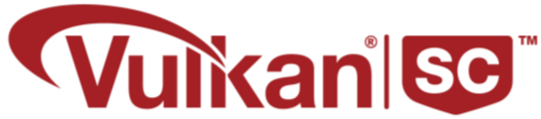
5. The Implementer attests that the Implementation executed on the Reference System suitably represents the Implementation on the Target System and the Implementer believes CTS would execute with comparable results on the Target System.

Conformance may be claimed for any component-level products necessary for a Conformant Implementation.

P10. Mark and Logo Usage Guidelines

The following usage guidelines must be followed for any use of the Marks identified below:

- The Vulkan SC Mark and Logo may be used only with respect to Conformant Products that have passed the Vulkan SC Conformance Process.
- "®" must be used as shown below with the first use of the written Mark in a document.
- Each document that uses the Marks must include the following text: "Vulkan SC and the Vulkan SC logo are registered trademarks of the Khronos Group Inc." The Marks are:
- Vulkan SC®
- The Vulkan SC Logo



`