

# 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 - November 2007** – corrected bug fix versioning numbering in Test Code Source Package, clarified OpenGL ES fees
- **V2 – February 2008** – updates to OpenKODE submission package in Attachment D

## 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;

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

**“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;

**“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 the latest version of the Specification for which an Adopter has executed an Adopters Agreement;

**“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;

**“Review Notice”** means a written notice sent to an Adopter to identify 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 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;

**“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 a Conformance Fee 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.

Executing the Adopters Agreement sets the Adopters Paid Specification Version at the version of Specification defined in the Adopters Agreement and there is no limit on the number of Submissions 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.

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 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 web-site 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.

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 Khronos Board of Promoters.

## **5. Test and Process Updates**

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

The Adopters Package will contain all previously released versions of the Tests, Process Document and Specification Attachments for all released versions of the Specification. Adopters may use any version of the Tests but are strongly encouraged to use the latest version. The release dates of the Process Document and Specification Attachment used with the Tests must show that all three were the current version at some point in time, with all three release dates noted in the Submission.

## **6. Conformance Test Source and Porting**

Adopters must always use the latest version of the Adopters Package for all Submissions, Submission Updates and Submission Corrections.

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. In the case of potential bugs the Adopter must change the test source, submit a Waiver Statement for each potential bug and include any changed source code.

## **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 the Submission Package PLUS the following information (if not already included in the Submission Package):

- The version of the Specification being tested;
- The release date of the Tarball, Tests, Process Document and Specification Attachment that were used;
- A statement of conformance in which the Adopter lists all Conformant Products that are certified by the Adopter to be covered by this Implementation;
- All bugs and associated Waiver Statements – that 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. Submission Updates, Corrections and Deletions**

A Successful Submission may not be deleted or replaced. Adopters can make a Submission Update to add Conformant Products associated with a Conformant Implementation. All Submission Updates undergo the same review process as a Submission. Any previously updated Submissions shall continue to be available on the Submission Repository, but marked as superseded, once a Submission Update is successful.

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

## **9. Submission Review Process**

All Submissions 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 too:

- 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.

Khronos shall appoint a Review Board 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 Board shall take all available information under consideration and determine by majority vote (50% or more of the Review Board members) whether the Submission needs a Correction. If the Review Board takes more than 30 days to reach a decision after the Review Notice is raised then the Submission or Submission Update shall be deemed successful;
- If the Submitter feels that the Review Notice is being incorrectly applied it may request the Khronos Board of Promoters to make a final decision, through its normal voting process, on the validity of the Submission.

After Submission Corrections have been made to resolve any Review Notices and the Review Period has passed with no further Review Notices being raised the Submission is deemed to be a Successful Submission. Entirely at its own discretion, the Review Board may waive the remainder of the Review Period once there are no outstanding Review Notices 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.

## 10. 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.

## 11. Suggestions

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 Specification Working Group that shall take any appropriate action, entirely at their own discretion.

## 12. Press Release Wording Guidelines

Statements that may always be used by Khronos members include:

- Adopter supports the Specification;
- Adopter encourages developers to use the Specification.

Statements that may be used by an Adopter that has a Submission in its Review Period:

- Adopter's Product is designed to implement the Specification;
- Adopter has submitted Product to the Khronos Conformance Process;
- Adopter expects Product to successfully pass the Khronos Conformance Process;
- If Adopters product passes the Khronos Conformance Process it will become officially certified to use the Marks.

If any of the last four statements are used - then the following footnote text should also be used:

- "Product is based on the published Khronos Specification specification, but only successfully passing products are licensed by Khronos or other parties to use the Specification trademark. Current conformance status can be found at [www.khronos.org](http://www.khronos.org)."

# Attachment A

## OpenGL ES Conformance Process Details

### A1. Change History and Version

- February 2007 – first version attached to generic process document

### A2. Paid Specification Versions Covered

Any version of OpenGL ES 1.X up to OpenGL ES 1.1 including minor release updates, and any version of OpenGL ES 2.X up to OpenGL ES 2.0 including minor release updates (i.e. OpenGL ES 1.1.1 and OpenGL ES 2.0.1 etc. are covered).

### A3. Conformance Fees

If an Adopter paid OpenGL ES 1.0 Conformance Fees under a previous version of the Process then these fees may be discounted from Conformance Fee. The Conformance Fee for each version of the Specification is below – for Adopters that are not Contributors or Promoters of Khronos – these fees shall be increased by \$5K:

	Conformance Fee	Upgrade Conformance Fee
<b>OpenGL ES 1.X up to OpenGL ES 1.1</b>	\$10K	From 1.0: \$10K - minus paid 1.0 Fees
<b>OpenGL ES 1.X and OpenGL ES 2.X up to OpenGL ES 2.0</b>	\$14K	From 1.0: \$14K - minus paid 1.0 Fees From 1.1: \$8K

### A4. 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>.

### A5. 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 gunzip 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.

## A6. 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.

## A7. Review Period

30 Days

## A8. Conformant Product Criteria

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

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.

## A9. 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 ES logo is a trademark of Silicon Graphics Inc. used by permission by Khronos."

(i) OpenGL® ES

(ii) OpenGL ES Logo:



# 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 Contributors or Promoters 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

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

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:



# Attachment C

## OpenVG Conformance Process Details

### C1. Change History and Version

- February 2007 – first release

### C2. Paid Specification Versions Covered

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

### C3. Conformance Fees

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

	Conformance Fee	Upgrade Conformance Fee
OpenVG 1.0	\$10K	NA

### 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 – 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

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

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 Contributors or Promoters of Khronos – these fees shall be increased by \$5K:

	Conformance Fee	Upgrade Conformance Fee
<b>OpenKODE 1.0 Provisional (any revision) and OpenKODE 1.0</b>	\$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.