ISO/IEC JTC 1/SC 29/WG 03 N1721



**ISO/IEC JTC 1/SC 29/WG 03  
MPEG Systems   
Convenorship: KATS (Korea, Republic of)**

**Document type:** Output Document

**Title:** Workplan for the reference software and conformance for ISO/IEC 14496-34

**Status:** Approved

**Date of document:** 2026-01-23

**Source:** ISO/IEC JTC 1/SC 29/WG 03

**No. of pages:** 7 (with cover page)

**Email of Convenor:** young.L @ samsung . com

**Committee URL:** <https://isotc.iso.org/livelink/livelink/open/jtc1sc29wg3>

**INTERNATIONAL ORGANIZATION FOR STANDARDIZATION**

**ORGANISATION INTERNATIONALE DE NORMALISATION**

**ISO/IEC JTC 1/SC 29/WG 03 MPEG SYSTEMS**

**ISO/IEC JTC 1/SC 29/WG 03 N1721**

**January 2026, online**

|  |  |
| --- | --- |
| **Title** | **Workplan for the reference software and conformance for ISO/IEC 14496-34** |
| **Source** | **WG 03, MPEG Systems** |
| **Status** | **Approved** |
| **Serial Number** | **25910** |

Table of Contents

[ISO/IEC JTC 1/SC 29/WG 03 N1721 1](#_Toc220094418)

[1 Introduction 2](#_Toc220094419)

[2 Overview of tools 2](#_Toc220094420)

[3 Work plan 3](#_Toc220094421)

[4 Utility software management 4](#_Toc220094422)

[4.1 Software coordinators 4](#_Toc220094423)

[4.2 Procedures 4](#_Toc220094424)

[4.3 Software projects 5](#_Toc220094425)

[4.3.1 General 5](#_Toc220094426)

[4.3.2 Word2SDL 5](#_Toc220094427)

[4.3.3 SDLValidator 5](#_Toc220094428)

[5 Conformance software management 5](#_Toc220094429)

[5.1 Software coordinators 5](#_Toc220094430)

[5.2 Procedures 5](#_Toc220094431)

[5.2.1 Branches 5](#_Toc220094432)

[5.2.2 Tagging 6](#_Toc220094433)

[5.3 Software projects 6](#_Toc220094434)

[5.4 General 6](#_Toc220094435)

[5.4.1 libSDL 6](#_Toc220094436)

# Introduction

ISO/IEC 14996 part 34 Syntactic description language (SDL) formalizes the syntactic description language for describing the structure of binary data used throughout other MPEG standards. It covers the representation of the SDL specification in plain text, the syntax of the SDL and the semantic rules of the SDL.

This document serves as project management tool to organize and track the progress of developing a series of tools related to SDL. Some of the developed tools are utility software whose purpose is to assist in editing SDL declaration while other tools are conformance software of ISO/IEC 14496-34.

# Overview of tools

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tool name | Repository location | Type | Status | Description |
| libSDL | [MPEG / Systems / SDL / libSDL · GitLab](https://git.mpeg.expert/MPEG/Systems/sdl/libsdl) | Conformance | Work-In-Progress | Library to parse SDL specifications. |
| SDLValidator | [MPEG / Systems / SDL / SDLValidator · GitLab](https://git.mpeg.expert/MPEG/Systems/sdl/sdlvalidator) | Utility | First version done | Command-line tool to validate SDL specifications. |
| Word2SDL | [MPEG / Systems / SDL / Word2SDL · GitLab](https://git.mpeg.expert/MPEG/Systems/sdl/word2sdl) | Utility | Done | Extract SDL declarations from Word document to plain text. |
| mpeg-sdl-parser | [MPEGGroup/mpeg-sdl-parser: MPEG SDL parser implemented in TypeScript.](https://github.com/MPEGGroup/mpeg-sdl-parser) | Conformance | First version done against 1ed | Command-line tool to validate SDL specifications. |
| mpeg-sdl-editor | [MPEGGroup/mpeg-sdl-editor: MPEG SDL web based editor.](https://github.com/MPEGGroup/mpeg-sdl-editor) | Utility | Done | Web editor for SDL. |
| SDLCompiler (tentative) | todo | Utility | Not started | Generate source code to read and write binary data based on its SDL declaration. |

# Work plan

|  |  |  |  |
| --- | --- | --- | --- |
| Milestones | Dates | Goals | Status |
| MPEG #149 | 24th January 2025 | * Create GitLab project for Word2SDL, libSDL and SDLValidator. | Agree on Workplan and procedure |
| AhG call | March 2025 *(date will be announced on the reflector)* | * Complete Word2SDL * Complete libSDL and SDLValidator. |  |
| MPEG #150 | 31st March 2025 | * Submit as input a diff text between current ISBOMFF classes vs. corrected ISOBMFF classes. | To complex to do in one effort the entire specification, postponed. |
|  |  | * Submit as input a report on how to fix each type of issues (e.g. ‘avc1’). | The impact analysis has been submitted: [#112](https://git.mpeg.expert/MPEG/Systems/sdl/contributions/-/issues/112), [m72315](https://dms.mpeg.expert/doc_end_user/documents/150_OnLine/wg11/m72315-v1-m72315.zip). |
|  |  | * Complete libSDL and SDLValidator. | [SDLValidator](https://git.mpeg.expert/MPEG/Systems/sdl/sdlvalidator) can validate a SDL file with a list of classes from a syntactic point of view.  Note: Needs to be pushed on MPEG GitLab |
| AhG call #1 | May  (to be announced) | * Push [SDLValidator](https://git.mpeg.expert/MPEG/Systems/sdl/sdlvalidator) to MPEG GitLab. * Progress libSDL and SDLValidator. * Major topic will be TuC review. |  |
| AhG call #2 | June (to be announced) | * Complete libSDL and SDLValidator. |  |
| MPEG #151 | 30th June 2025 | * Submit a proposed full ISBOMFF SDL declaration with proper syntax and easy reading. |  |
| AhG call #1 | 9th September 2025 | * Review progress on conformance tools * Define task for upcoming meeting. |  |
| MPEG #152 | 7th October 2025 | * Progress on conformance tools development. | Bug fixes and new features in the SDL web editor for friendlier editing an error reporting. |
| AhG call #1 | 4th November 2025 | * Check conformance work progress. * Review N1662. * (non conformance) Review input of MPEG #152 | xf |
| AhG call #1 | 9th December 2025 | * Check conformance work progress. * (non conformance) Review input of MPEG #152 |  |
| File format AhG call | Did not happen | * Present N1662 on the ISOBMFF syntax fixes |  |
| MPEG #153 | 19th January 2026 | * Progress conformance work | Progression happens asynchronously with MPEG meeting |
| MPEG #154 | 27th April 2026 | * Support FileFormat BoG to use SDL tools * Assess current status of tools and next steps |  |

# Utility software management

## Software coordinators

The software coordinators are:

* Emmanuel Thomas (thomase@xiaomi.com)
* Nick Ryan (nick.ryan@hoot.works)

## Procedures

The software development of the repositories are done based on a merge request process. Contributors including software coordinators create their own fork and provide input via merge requests against the main branch of each project.

## Software projects

### General

The set of tools developed for conformance software are designed in a way to be able to evolve as new editions and amendments are created.

### Word2SDL

A script to extract from a Word document the SDL classes found into a plain text file.

### SDLValidator

Command line tool to check syntax and semantics of an SDL specification stored in a text file. Uses the libSDL from the SDL conformance software package.

# Conformance software management

## Software coordinators

The software coordinators are:

* Emmanuel Thomas (thomase@xiaomi.com)
* Nick Ryan (nick.ryan@hoot.works)

## Procedures

### Branches

Conformance software repository can contain three types of the branch main, project and staging branches as depicted in .

|  |  |  |  |
| --- | --- | --- | --- |
| Type of branch | Number of branches | Naming pattern | Protected to push |
| Main | One | main | Yes |
| Project | One per ISO project | project-(AMD*n*)-ED*n* | Yes |
| Staging | One per ISO project | staging-(AMD*n*)-ED*n* | No |

Each repository has a unique branch called main. This branch is synced with the latest published edition of the corresponding specification. If there is no yet published edition, this branch is empty.

The project branches are created for each ISO project (new edition or amendments), possibly before that the ISO project was effectively requested. For example, a WD can start before a new edition or an amendment has been requested to ISO. In this case, the project branch is created ahead of the existence of the ISO project. On the project branches, only merge request from the staging branches by the software coordinators are allowed.

The staging branches are short-lived. They purposes is to collect the contributions submitted at a given meeting. For each ISO project, there should be at most one staging branch with the naming tag referring to its corresponding project branch. Contributors provide contributions via merge requests against the appropriate project staging branch.

After the meeting, the software coordinators combine all the received merge requests in the staging branches and after validation, the staging branch is merged into the associated project branch and tagged appropriately.

### Tagging

In the project and main branches, commits resulting from a new milestone are tagged with the following tags:

* Working Draft with “WD”
* Committee Draft with “CD”
* Draft International Standard with “DIS”
* Final Draft International Standard with “FDIS”
* International Standard with “IS”

NOTE There may be some more prefixes needed for example for potential improvement of revised text. Those will be defined when needed.

## Software projects

## General

The set of tools developed for conformance software are designed to evolve as new editions and amendments are standardized.

### libSDL

Typescript library implementing syntactic parser and outputting an abstract syntax tree. Visitor framework for performing semantic checks of the abstract syntax tree.