

**INTERNATIONAL ORGANISATION FOR STANDARDISATION  
ORGANISATION INTERNATIONALE DE NORMALISATION  
ISO/IEC JTC1/SC29/WG11  
CODING OF MOVING PICTURES AND AUDIO**

**ISO/IEC JTC1/SC29/WG11  
MPEG2002/M8932  
October 2002, Shanghai, China**

**Source: Input for MPEG 62<sup>nd</sup> Meeting in Shanghai**

**Status: Draft**

**Title: MPEG-4 Systems Reference Software Updates**

**Authors: Jean Le Feuvre, Cyril Concolato, Jean-Claude Dufourd**

**Organisation : ENST**

## **Introduction**

ENST is glad to release a new MPEG-4 2D player for the MS Windows® platform, code-name Osmose. This software was developed as a test-bed for MPEG-4 Advanced Text and Graphics Working Draft, and has been extended to support all 2D graphics features of MPEG-4 Systems. Along with this release come minor updates in MP4Enc, libisomp4 and DMIF MP4 file reader modules.

## **I Osmose Player**

### I.1 Overview

The Osmose player is based on the previous IM1 2D player from CSELT and uses roughly the same rendering architecture. The main difference is that graphics rendering is done through Gdiplus, a new graphics API from Microsoft introduced with Windows XP products line, thus only Windows platform with Gdiplus support can be used (Windows98, NT 4.0 and higher). Microsoft DirectX is also required on the system.

The audio architecture is unchanged but only support for DirectSound has been kept.

Notes on installation and compilation are included in the player README directory.

### I.2 Architecture

The Osmose player enhances the composition mechanisms of IM1-2D by allowing choosing among several rendering modes.

- *Direct Rendering Mode*: in this simple mode, the compositor draws the scene graph at each rendering iteration. This mode is less CPU intensive but can be

extremely graphics intensive in complex scenes. This mode is very useful as a debugging mode.

- *Bounds Tracking Mode Fixed-Rate*: in this more complex mode, the compositor tracks objects bounds as being displayed and computes the minimal area to be redrawn. This is less graphics-intensive but consumes more CPU. The graph rendering is performed at the required frame-rate.
- *Bounds Tracking Mode Free Rate*: Same as the previous mode except that frame rendering is triggered by changes in the scene graph state (BIFS commands/Anim and Routes) in order to perform the minimal amount of operations. This is the most efficient mode.

As said above the Osmose player is able to detect scene graph changes in order to decide whether rendering shall be performed or not. The IM1 Core has been slightly modified in order to support this feature by notifying the presenter:

- each time a system Access Unit is decoded (BIFS, BIFS-Anim, OD, OCI)
- each time one or more routes are activated

Please note that this is not completely optimized since some routes may replace a node field by the exact same value, in which case this is seen as a scene graph state change by the core even though it is not. Modifications to the Core could be made to optimize these situations in the case of basic types (SF/MF color, int, bool, float and time) in order to track whether a route has indeed changed target node or not. This is however a quite complex matter and it is not expected to be solved any time soon.

### I.3 New Features

#### a) *Osmose Player*

The following features are available in the Osmose player:

- Anti-Aliasing through Gdiplus
- Dynamic rendering mode switching
- Rendering can be set to high drawing speed or high graphics quality
- Scalable zoom can be used when changing window size
- Support for pixel metrics and float metrics in the player

## b) MPEG-4 2D Nodes

The Osmose Player aims at being a Complete2D MPEG-4 player but not all the features are available yet.

The following nodes are fully implemented in the player:

- Building block nodes: Group, OrderedGroup, Transform2D, Switch, Inline
- Appearance nodes: Material2D, lineProperties, ImageTexture, PixelTexture CompositeTexture2D, TextureTransform, MaterialKey (with bitmap only). Texture Mapping is implemented on all geometry nodes.
- Geometry nodes: Curve2D, Bitmap, Circle, IndexedLineSet2D, PointSet2D, Rectangle, Text.
- Sensor nodes: TimeSensor, TouchSensor, DiscSensor, PlaneSensor2D, ProximitySensor2D.
- Postprocessing nodes: Form, Layer2D.
- Other nodes: WorldInfo, Anchor, Background2D

The following nodes are partially implemented in the player:

- MovieTexture (the node works fine but movie looping isn't supported in IM1).
- MatteTexture (the node works fine, but only the effects implemented in the original player have been re-implemented, which makes the implementation not fully complete).
- IndexedFaceSet2D (texture modification through the texture coordinates field of IndexedFaceSet2D is not implemented)

The following nodes are implemented but not tested:

- Layout
- TemporalGroup, TemporalProxy.

**Note:** FlexTime is not currently supported due to errors in the original reference software which were altering the compositor behavior. The FlexTime code has been integrated in the Osmose compositor but is commented out since it breaks node rendering.

### I.3 BIFS-Animation

The Osmose player has been successfully compiled with support for BIFS-Anim, and the AnimationStream Node has been implemented in order to test BIFS-Anim. However the testing hasn't been extensive given the few bit-streams available. Some tests were made with Envivio's bit-streams from the MPEG4 Conformance Set, it appears that the stream is being decoded (no crash) but the results were quite different from our expectations. We didn't investigate any further given the few testing of BIFS-Anim in latest IM1 versions.

### I.4 Future

We believe Osmose player is stable enough to play most of 2D MPEG-4 contents and flexible enough to be used as a test-bed for advanced 2D features of MPEG-4 BIFS. Therefore we propose the Osmose player as a candidate for inclusion in ISO/IEC 14496-5.

## II Advanced Text and Graphics

As said above the original goal of developing Osmose was implementation of new nodes especially in the context of Advanced Text and Graphics effort currently happening in MPEG-4 Systems. The Osmose player integrates new nodes based on the current Working Draft (generated from beta v6 template available at [www.fy.com/MPEG4](http://www.fy.com/MPEG4)). We have implemented the following nodes:

- TransformMatrix2D (full implementation)
- Viewport (currently only supported in CompositeTexture2D with no fit/align support)
- XlineProperties (full implementation)

ATG WD also proposes amendment items on BIFS V1 nodes, namely Curve2D, Layout and CompositeTexture2D.

- The proposed functionality of Curve2D is implemented in Osmose, but the coding of the node isn't. The Curve2D type field accepts the new range of the enhancedType values, so that the new types can be tested without quantization by using the Curve2D.type field instead of the proposed Curve2D.enhancedType.
- CompositeTexture2D will be discussed in contribution **m8929** for the 62<sup>nd</sup> MPEG Meeting in Shanghai.
- As said above, Layout is currently not tested in Osmose and therefore neither are its extensions.

These specific features will be cleaned up when the template for BIFS V1 is updated accordingly.

## III ISO/IEC 14496-5 Updates

### III.1 User Input

The User Input module has been cleaned-up and reintegrated in the core, and is working in the Osmose player. The code is mature enough and we suggest merging the UI files (./UI directory) in the main core (./Core directory).

### III.2 MP4 Related

The MP4Enc module has been updated to support empty streams (used for ESD URL and UI streams). The libisomedia has been updated to support this feature.

Note: Apple Computer has just released a new version of the libisomedia, but this version is currently not on CVS. The CVS version only will be updated to support the MP4Enc updates.

The MP4 file reader has been updated to signal End Of Stream to the core.

### III.3 Main Core System

There are several updates in the core module:

- Extension of the base Application class, the virtual Core interface to the user application
- Extensions to Executive / PresenterBase to signal scene graph state changes
- Not backward compatible changes in Capabilities.h file. This means that all IM1 decoders shall be recompiled with the new Capabilities.h in order to run in the Osmose player (the MediaFormat enumeration has changed).
- The IM1-2D player has been fixed to support latest changes in UserInput and Core module introduced during Osmose development.

### III.4 IM1 decoders

A new decoder for PNG textures has been implemented, available in IM1Decoders/PNG. Compiled (Win32) static libraries for PNG and ZLib are included as well as copyright notice. PNG streams are defined in the following way:

- No decoder specific configuration
- StreamType is visual
- ObjectTypeIndication is 0x6D

To produce MPEG-4 content using PNG with Mux or MP4Enc tools, set the muxInfo.streamFormat field of the desired stream in the script file to JPEG and specify an ObjectTypeIndication of 0x6D. Multiple PNG files cannot be concatenated into a single stream.