

# Deliverable

Project Acronym:	ImmersiaTV
Grant Agreement number:	688619
Project Title:	<i>Immersive Experiences around TV, an integrated toolset for the production and distribution of immersive and interactive content across devices.</i>

## D3.6 Interaction and Display

**Revision:** 0.9

**Authors:**

Szymon Malewski, PSNC

Isaac Fraile, i2CAT

Ibai Jurado, i2CAT

**Delivery date:** M24

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement 688619

Dissemination Level

P	Public	x
C	Confidential, only for members of the consortium and the Commission Services	

**Abstract:** This deliverable describes the software components for end-user devices delivered for Pilot 1 and Pilot 2.

## REVISION HISTORY

Revision	Date	Author	Organisation	Description
0.1	20/12/2016	Szymon Malewski	PSNC	Document edition
0.2	09/01/2017	Joan Llobera	i2CAT	Final Review
0.3	10/01/2017	Szymon Malewski	PSNC	Final formatting
0.4	02/06/2017	Szymon Malewski	PSNC	Revision after mid-term review
0.5	06/09/2017	Szymon Malewski	PSNC	Update for pilot 2
0.6	02/10/2017	Isaac Fraile	i2CAT	Web player description
0.7	11/10/2017	Szymon Malewski	PSNC	Final formatting after review
0.8	13/10/2017	Maciej Glowiak	PSNC	Final corrections
0.9	22/12/2017	Juan Antonio Núñez	i2CAT	Document review

### Disclaimer

The information, documentation and figures available in this deliverable, is written by the **ImmersiaTV** (*Immersive Experiences around TV, an integrated toolset for the production and distribution of immersive and interactive content across devices*) – project consortium under EC grant agreement H2020 - ICT15 688619 and does not necessarily reflect the views of the European Commission. The European Commission is not liable for any use that may be made of the information contained herein.

### Statement of originality:

This document contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both.

## CONTRIBUTORS

---

First Name	Last Name	Company	e-Mail
Joan	Llobera	I2cat	<a href="mailto:Joan.llobera@i2cat.net">Joan.llobera@i2cat.net</a>
Szymon	Malewski	PSNC	<a href="mailto:szymonm@man.poznan.pl">szymonm@man.poznan.pl</a>
Isaac	Fraile	I2cat	<a href="mailto:Isaac.fraile@i2cat.net">Isaac.fraile@i2cat.net</a>
Juan Antonio	Nunez	I2cat	<a href="mailto:juan.antonio.nunez@i2cat.net">juan.antonio.nunez@i2cat.net</a>

## CONTENTS

---

Revision History .....	1
Contributors .....	2
Table of Figures.....	4
List of acronyms .....	5
1. Introduction .....	6
2. Architecture and functionality.....	6
2.1. Player .....	6
2.2. Synchronization .....	8
3. Installation and usage guide .....	10
3.1. SessionManager .....	10
3.2. Unity Player .....	10
3.3. Web Player .....	12
4. Code documentation.....	15

## TABLE OF FIGURES

---

Figure 2-1: Players interacting with session manager .....	6
Figure 2-2: Unity player general architecture .....	7
Figure 2-3: Basic diagram of Session Manager - Player interaction .....	9
Figure 3-1: Session manager window .....	10
Figure 3-2: Content selection screen .....	11
Figure 3-3: Viewing format selection screen.....	11
Figure 3-4: Started content in TV format .....	12
Figure 3-5: Web player main menu .....	13
Figure 3-6: Web player TV mode with the QR codes.....	14
Figure 3-7: Web player on table mode .....	14

## LIST OF ACRONYMS

---

Acronym	Description
AVC	Advanced Video Codec
DASH	Dynamic Adaptive Streaming over HTTP
DVB-CSS	Digital Video Broadcasting – Companion Screen Synchronization
HMD	Head mounted display
RTSP	Real Time Streaming Protocol
VOD	Video on demand
QoE	Quality of Experience

## 1. INTRODUCTION

---

The display component is responsible for integrating the video, audio and data streams in a coherent omnidirectional scene adapted to immersive displays and second screens. It supports interaction, parsing the user input (head movements, tablet moved around, finger gestures) and adapting the environment appropriately to the reactions expected. It also allows to synchronize end-user devices and second screens using a multimedia server to orchestrate the different video streams.

## 2. ARCHITECTURE AND FUNCTIONALITY

---

The display component is a multi-platform software solution that is able to:

- synchronize and integrate different video and audio streams in a consistent experience,
- handle the interaction specific to each device – looking around the scene by
  - head movements for the HMD,
  - device movements or finger gestures for the tablet,
- synchronize content at the frame level with other displays in the local network

The architecture of the display component contains two kinds of connected devices: receiver devices (TV Set, HMD, Tablet) and a session management device (see Figure 2-1).

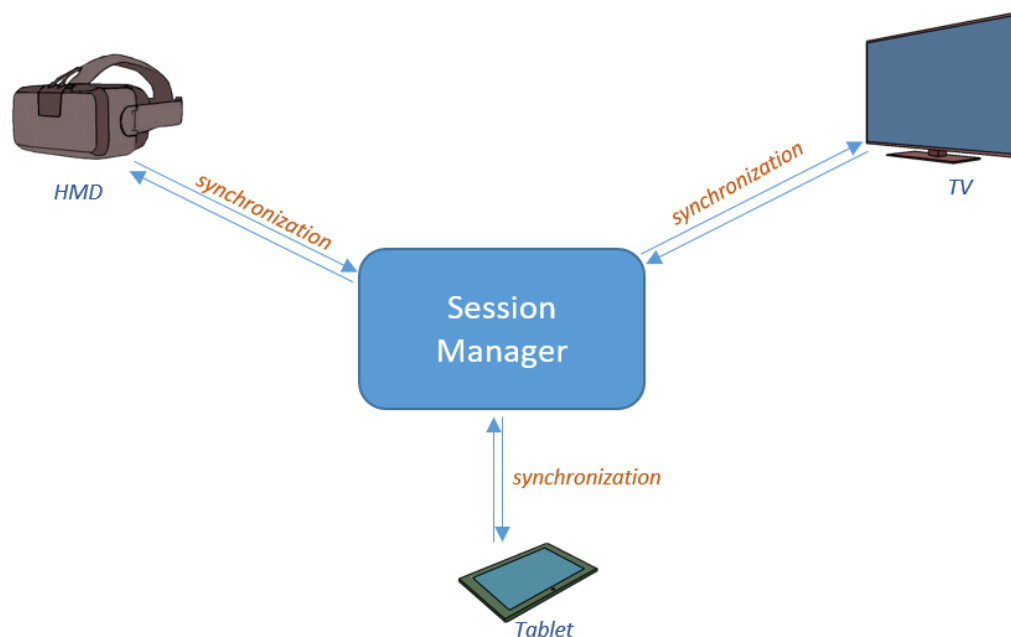


Figure 2-1: Players interacting with session manager

### 2.1. Player

The receiver device runs the ImmersiaTV interaction and display software (in short, the ImmersiaTV player). This software is targeting the general consumer. It was tested with several devices:

- NVIDIA SHIELD Android TV
- Samsung Gear VR with Samsung Galaxy S6, S7, S8

- Tablet Samsung Galaxy Tab S
- Tablet Asus Nexus
- Tablet Dell Venue 11 Pro
- Tablet Google Pixel C
- HTC Vive
- Oculus Rift

Two versions of player were implemented:

- Player based on the Unity3D engine, currently available for Windows and Android and adapting the experience to the particular characteristics of each device. The processing of the media streams is performed using the GStreamer open-source framework. It receives and decodes different audio and video streams and delivers resulting frames to Unity3D for rendering. This workflow is presented on Figure 2-2.

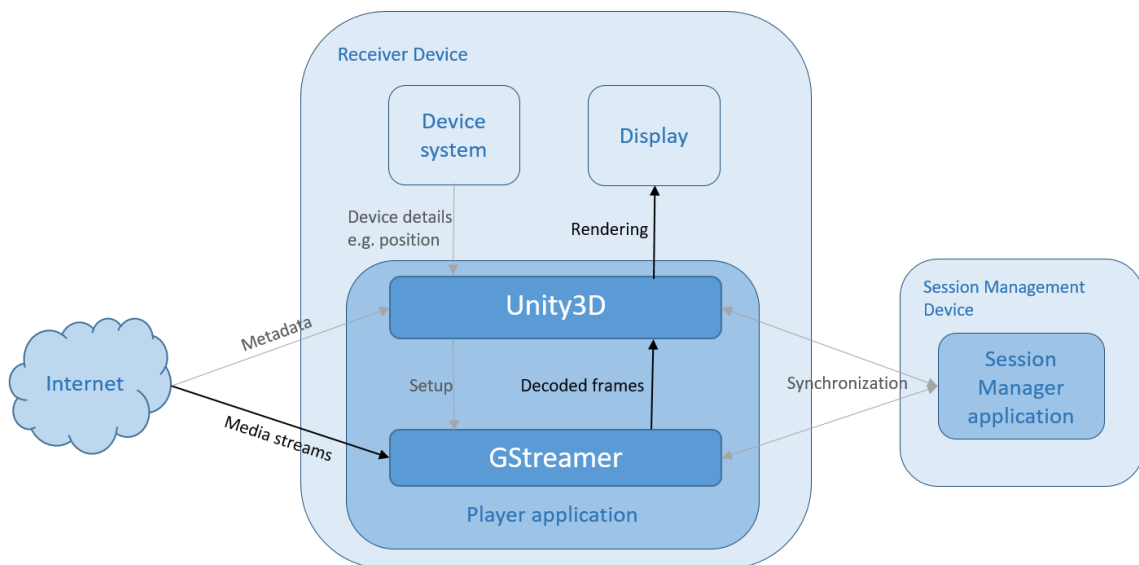


Figure 2-2: Unity player general architecture

- Web player allowing deployment on a wide variety of end-user devices. It was implemented using web technologies such as Javascript and HTML5. For playback of the media streams in players, it has used the open source Javascript library Dash.js version 2.5 provided by the DASH Industry Forum that enables to reproduce MPEG-DASH streams. To create and compose the scenes it has used the open source Javascript library Three.js, which uses WebGL to display the content and enable the possible interactivity with the user. To finish it has used the Omnitone.js library that use the Web Audio API to add support to ambisonic decoding.

## Key features

Basic functionality of players includes:

- Content selection – player presents list of content available on server (see D3.5).
- Playout of ImmersiaTV content (defined by metadata created in ImmersiaTV production tools):
  - Selecting video timeline based on device type.
  - Presenting 360° scene on HMD and mobile devices.
  - Scene can be composed form multiple video streams (portals).
  - Possible interactivity with the user (opening/closing portals, selecting camera).
  - Video transitions on user actions.



- Support for video streams:
  - Encoding: AVC
  - Resolution: up to 4K (depends on device)
  - Projection: equirectangular or cubic
  - Streaming protocol: DASH, RTSP
- Support for images. (PNG, JPEG)
- Support for VOD (pilot 1) and live (pilot 2) content.
- Synchronization with other devices (see next paragraph).
- Logging QoE data (device parameters, displayed content id, delays, looking direction).
- Support for spatial audio.

## 2.2. Synchronization

The session management device is connected to the same local network as the players and coordinates the distributed playback experience. Its main task is to make sure that all players synchronize to the same clock and get appropriate content.

Session manager can be a standalone application independent from other players, running on a separate Windows or Linux PC (for test and demo scenarios) or integrated with the player application, so any player can act as session manager, removing the need for an additional device on the network.

For the synchronization, ImmersiaTV player implements emerging broadcast standard DVB-CSS (Companion Screen Synchronization), that is going to be a part of HbbTV 2.0. Especially it uses protocols defined by the standard:

- Discovery – for finding and connecting to session manager in local network,
- Wallclock (DVB-CSS-WC) – to synchronize clocks of all connected devices, to have common reference time,
- Content Information and Identification (DVB-CSS-CII) – for sharing the information about currently played content,
- Timeline Synchronization (DVB-CSS-TS) – for coordinating playout of the content.

Below Figure 2-3 presents general diagram of synchronization process.

The use of DVB-CSS-WC in web environments is problematic, due to the fact that these synchronization protocol is based on UDP sockets, for which current web browsers do not provide support. In order to solve that, a new service has been implemented. This new service slightly modify the DVB-CSS-WC protocol, using websockets instead of UDP sockets to establish the common WC.

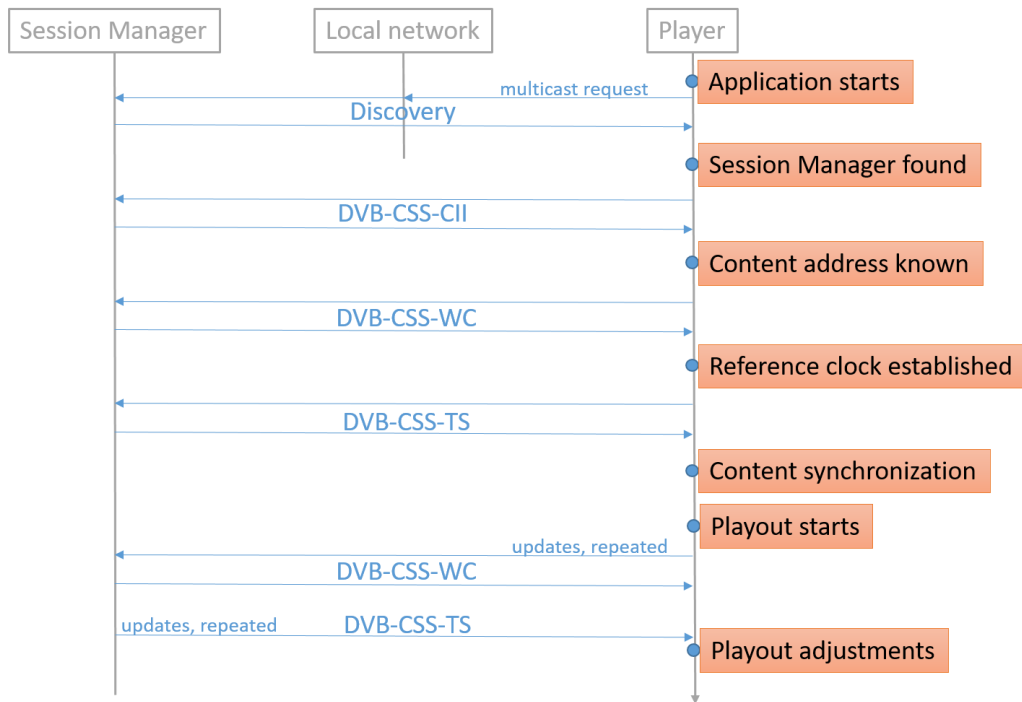


Figure 2-3: Basic diagram of Session Manager - Player interaction

## 3. INSTALLATION AND USAGE GUIDE

### 3.1. SessionManager

Download <ftp://ftp.immersiatv.eu/releases/0.19/sessionmanager.zip> and extract its content.

Start session-manager.exe. It will open console window (see figure 3) and will provide synchronization for ImmersiaTV players started in the local network.

```
2017-09-06 12:04:24,774 [19] INFO ImmersiaTV.SessionManager.ContentIdentificationSession - New client connected 10.0.75.1:8442
=====
Discovery enabled on address: 10.0.75.1
DVB-CSS-CII listening on address:
ws://10.0.75.1:4649/dvbcscii
and returns:
{
  "protocolVersion": "1.1",
  "wcUrl": "udp://10.0.75.1:5000/",
  "tsUrl": "ws://10.0.75.1:4649/dvbcscsts",
  "private": [
    {
      "contentSetterUrl": "ws://10.0.75.1:4649/cs",
      "videoControllerUrl": "ws://10.0.75.1:4649/vc",
      "gstClockUrl": "udp://10.0.75.1:5001",
      "wsClockUrl": "ws://10.0.75.1:4649/wc",
      "type": "tag:url"
    }
  ]
}
=====
2017-09-06 12:04:24,914 [10] INFO ImmersiaTV.SessionManager.ContentIdentificationSession - Client disconnected.
Controls: [Esc] - quit, [Enter] - reset content
```

Figure 3-1: Session manager window

### 3.2. Unity Player

- To install and start Unity player download a version suitable for your platform from <ftp://ftp.immersiatv.eu/releases/0.19/players/> and follow instructions:

<b>PC</b>	<ul style="list-style-type: none"> <li>Create a folder in your hard drive</li> <li>Unzip win64.zip in the created folder</li> <li>Execute immersiaTV_Player.exe</li> </ul>
<b>PC with Oculus Rift</b>	The process is similar to the PC. See above instructions, using win64wOculus.zip for this case.
<b>PC with HTC VIVE</b>	The process is similar to the PC. See above instructions, using win64wHTCVive.zip for this case.
<b>Android Mobile/ Cardboard</b>	<p>There are different ways to install an apk into an android device.</p> <ul style="list-style-type: none"> <li>Create a folder in your hard drive</li> <li>Copy ImmersiaTV_Android.apk in the created folder</li> <li>Connect your android device to the computer</li> <li>Move the apk file into \Download folder of your device storage</li> <li>Disconnect your device from the computer</li> <li>Execute the apk file from the device (use a File navigation app like FileManager)</li> </ul> <p>Once it is installed, you can remove the apk file from the mobile.</p>

- After application is launched, it automatically connects to the Session Manager started in the local network. If content have already been selected on another device application jumps directly to next point. Otherwise content selection screen is presented (See figure 4). Click the button with the content name to select it.

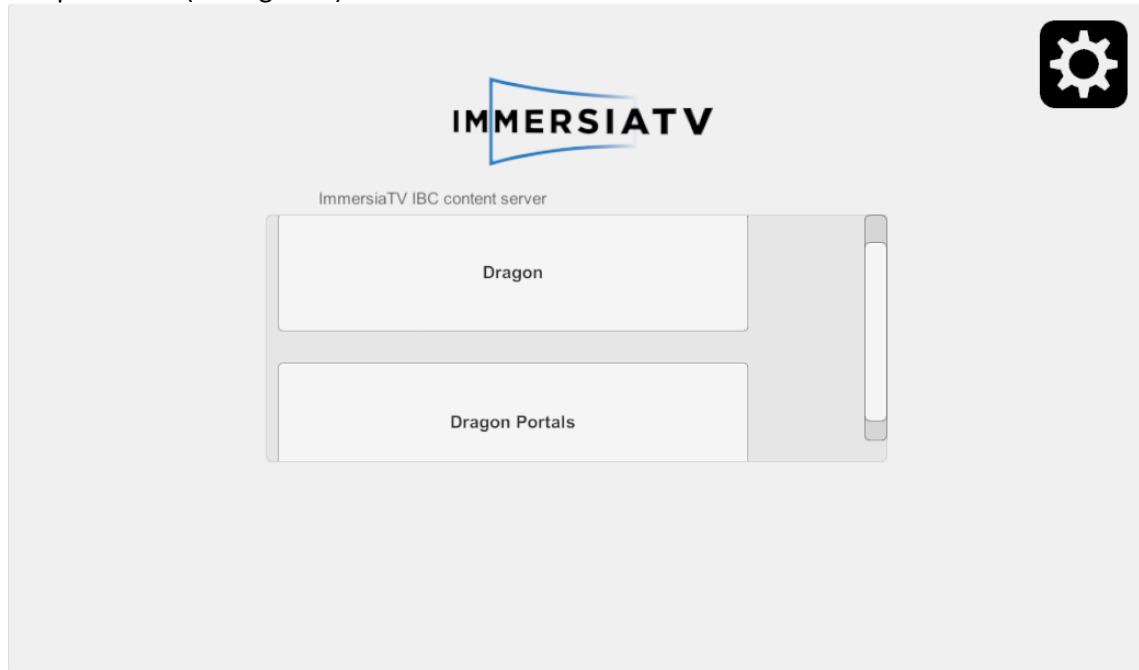


Figure 3-2: Content selection screen

- Next select viewing format – HMD, TV or Tablet (See figure 5). You can go back to content selection by clicking 'Back' button. Application versions for Samsung Gear VR, Oculus Rift and HTC Vive omits this screen and start playout directly on the HMD scene. On Android TV application starts on TV scene..

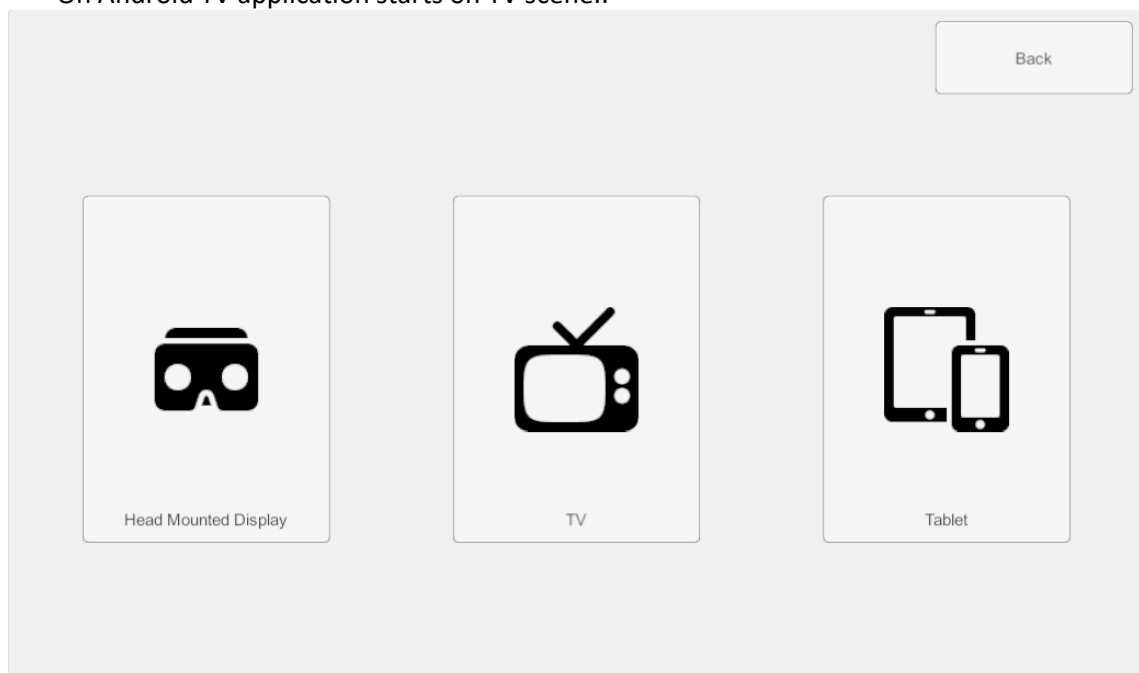


Figure 3-3: Viewing format selection screen

- Playout of selected content in desired format starts (See figure 6). It is synchronized between devices in the local network. HMD and Tablet formats allow to look around the scene and interact with portals.



Figure 3-4: Started content in TV format

- Pressing ESC key on PC, Oculus and HTC Vive players it will go back to the content selection screen. Pressing back button on Android devices it will go back to the content selection screen. In both cases, it does not affect the navigation that there is already a selected content.
- The interaction inside the different players changes to the technology used. <insert here the examples an uses of hmd pointing, mouse and touch>

### 3.3. Web Player

The web player does not need to be installed, it can be launched on Chrome or Firefox browsers through a URL or scanning a QR code. It is only necessary to publish on an http or https server such as an nginx or Apache Tomcat a suitable version of player from [ftp://ftp.immersiatv.eu/releases/0.21/player\\_web](ftp://ftp.immersiatv.eu/releases/0.21/player_web).



Figure 3-5: Web player main menu

Once the player is launched, it auto connects to the Session Manager. Users can see in the upper right part of the home screen whether the connection to the server has been successful or not through a pilot that changes colour to green if the connection has been successfully established or changes to colour red if not the connection could be established. If content have already been selected, it appears framed in colour yellow on the main screen.

In the main screen, the users can also select the device from which they want to play the content, such as table, TV or HMD clicking in the corresponding image of the bottom of the main screen.

To play the content is only necessary click on one of the listed elements.



Figure 3-6: Web player TV mode with the QR codes

As presented in the figure 3-6, in TV mode there are some QR codes that can be used to open directly the web player on other devices. Also, TV mode contains a control bar to do play/pause or seeks.

In the other two modes, HMD and tablet, the user can interact with the different elements of the scene. Through these interactions, the user can for instance, change between scenes or activate a transition to make an element appear or disappear.



Figure 3-7: Web player on table mode

Just like in the Unity player, the web player also shows its contents synchronized between connected devices.

- **TOOLS**

- **Unity Player**

The making of the player builds requires a lot of time. Setting up needs to differentiate the configuration for the different devices. So for this task of automatization it has developed a build maker script inside Unity3D engine.

## 4. CODE DOCUMENTATION

---

The code is documented using XML comments structure and parsed into html using DoxyGen (<ftp://immersiatv@ftp.immersiatv.eu/releases/0.19/documentation/unity-player-code/html/>). Doxygen supports most of the XML commands that are typically used in C# code comments. The XML tags are defined in Appendix E of the ECMA-334 standard (<http://www.ecma-international.org/publications/standards/Ecma-334.htm>), which defines the C# language used in the Unity Player.

The current release does include documentation in form of README file with the install instructions (<ftp://immersiatv@ftp.immersiatv.eu/releases/0.19/players/INSTALL.md>) and changelog file with the record of all notable changes made to the ImmersiaTV project ([ftp://immersiatv@ftp.immersiatv.eu/releases/0.19/players/CHANGELOG\\_PLAYER.md](ftp://immersiatv@ftp.immersiatv.eu/releases/0.19/players/CHANGELOG_PLAYER.md)).

Future versions will include a more detailed and structured documentation online.