

|  |  |
| --- | --- |
|  | **imotion** |
|  |  |
|  | Infomedia Documentation |
|  |  |
|  | Configuration |
|  |  |
|  | Technical Description |
|  |  |
|  | Version: 2.0 |
|  | State: Released |
|  | Classification: Internal use only |
|  | Author: RAN |
|  | Creation date: 2014-06-25 |
|  | Repository: $/Gorba/Main/Motion/Infomedia/Documents/ TD\_InfomediaDocumentation.docx |
|  | Gorba AG  Sandackerstrasse  9245 Oberbüren  Switzerland |

**Table of contents**

[1 Introduction 4](#_Toc403367517)

[1.1 Infomedia Version 4](#_Toc403367518)

[1.2 Purpose 4](#_Toc403367519)

[1.3 Intended Audience 4](#_Toc403367520)

[2 System Overview 4](#_Toc403367521)

[3 Components of Infomedia 5](#_Toc403367522)

[3.1 Composer 5](#_Toc403367523)

[3.1.1 Requirements to run Composer 5](#_Toc403367524)

[3.1.2 Interaction with other applications 5](#_Toc403367525)

[3.1.3 Shortcuts available 5](#_Toc403367526)

[3.2 DirectX Renderer 6](#_Toc403367527)

[3.2.1 Requirements to run DirectX Renderer 6](#_Toc403367528)

[3.2.2 Interaction with other applications 6](#_Toc403367529)

[3.2.3 Shortcuts available 6](#_Toc403367530)

[3.2.3.1 Alt+Enter 6](#_Toc403367531)

[3.2.3.2 F1 6](#_Toc403367532)

[3.3 Audio Renderer 7](#_Toc403367533)

[3.3.1 Requirements to run Audio Renderer 7](#_Toc403367534)

[3.3.2 Interaction with other applications 7](#_Toc403367535)

[3.3.3 Shortcuts available 7](#_Toc403367536)

[3.4 AHDLC Renderer 8](#_Toc403367537)

[3.4.1 Requirements to run AHDLC Renderer 8](#_Toc403367538)

[3.4.2 Interaction with other applications 8](#_Toc403367539)

[3.4.3 Shortcuts available 8](#_Toc403367540)

[4 Configuration 9](#_Toc403367541)

[4.1 Composer 9](#_Toc403367542)

[4.1.1 @XmlDoc(xsd=..\..\..\Common\Configuration\Source\Infomedia\Composer\Composer.xsd;xml=..\Source\Core\Composer.xml) 9](#_Toc403367543)

[4.2 DirectX Renderer 9](#_Toc403367544)

[4.2.1 @XmlDoc(xsd=..\..\..\Common\Configuration\Source\Infomedia\DirectXRenderer\DirectXRenderer.xsd;xml=..\Source\DirectXRenderer\DirectXRenderer.xml) 9](#_Toc403367545)

[4.3 Audio Renderer 10](#_Toc403367546)

[4.3.1 @XmlDoc(xsd=..\..\..\Common\Configuration\Source\Infomedia\AudioRenderer\AudioRenderer.xsd;xml=..\Source\AudioRenderer\AudioRenderer.xml) 10](#_Toc403367547)

[4.4 AHDLC Renderer 11](#_Toc403367548)

[4.4.1 @XmlDoc(xsd=..\..\..\Common\Configuration\Source\Infomedia\AhdlcRenderer\AhdlcRenderer.xsd;xml=..\Source\AhdlcRenderer\AhdlcRenderer.xml) 11](#_Toc403367549)

[5 Example usages 12](#_Toc403367550)

[5.1 DirectX Renderer example usages 12](#_Toc403367551)

[5.1.1 Define visible region on the screen example usage 12](#_Toc403367552)

[5.2 Audio Renderer example usages 12](#_Toc403367553)

[5.2.1 Speaker setup example usages 12](#_Toc403367554)

[5.2.2 Acapela example usage 13](#_Toc403367555)

[5.3 AHDLC Renderer example usages 14](#_Toc403367556)

[5.3.1 Signs example usage 14](#_Toc403367557)

[6 Glossary 15](#_Toc403367558)

**Modification management**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Version** | **Date** | **Name** | **Dept.** | **Modifications** | **State** |
| 0.1 | 2014-06-25 | RAN | SW | Initial version | draft |
| 1.1 | 2014-10-23 | RAN | SW | Added chapter 5 about example usages | draft |
| 1.2 | 2014-11-05 | RAN | SW | Updated minor changes in complete document based on DEL suggestions. Added chapter 5.2.2 | draft |

**Review**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date** | **Name** | **Dept.** | **Remarks** |
| 0.1 | 2014-06-26 | WES | SW | Reviewed, added glossary, minor formatting changes |
| 1.1 | 2014-10-30 | DEL | PH | Reviewed, |
| 1.2 | 2014-11-10 | WES | SW | Reviewed, application icons added. |

**Release**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date** | **Name** | **Dept.** | **Remarks** |
| 1.0 | 2014-06-26 | WES | SW | Release matching Infomedia 2.2.1426 |
|  |  |  |  |  |
| 2.0 | 2014-11-10 | WES | SW | Release matching Infomedia 2.4.1445 |
|  |  |  |  |  |

# Introduction

## Infomedia Version

This document covers Infomedia version **2.4.1445.**

## Purpose

The goal of this document is to describe the functionality and requirement of each component in Infomedia. It also describes in detail each parameter of every component in detail in order to configure it properly.

At the end of this document, the reader will have all the details to configure every component of Infomedia.

## Intended Audience

This document is addressed to product managers or customer project managers that are familiar with this product and are able to install, operate and maintain it.

# System Overview

Infomedia is a set of applications which handle the management of the complete presentation and the rendering of the information to different mediums. Each of the component acts as an individual application within the complete system. The components have to be configured individually and started by the System Manager as separate applications.

The various components of Infomedia are:

* Composer
* DirectX Renderer
* Audio Renderer
* AHDLC Renderer

The Composer is a compulsory component to be used when any information is to be presented to the end user. The different Renderers can be used both individually and together in a system based on the requirements. In the following chapters, each of the components is described along with the association to other components within Infomedia.

# Components of Infomedia

## Composer

The Composer application is part of Infomedia. It handles the management of the complete presentation.



### Requirements to run Composer

Composer must be configured to be started and handled by the System Manager. The Composer must have a separate folder for configuration under the “Config” folder and a separate folder for the binaries under “Progs” folder in the TFT system folder structure (refer to TD\_TFT2.0System document for details of the folder structure).

Composer requires the presentation file (\*.im2) and its associated files/folders to be present in the “Presentation” folder in the TFT system folder structure.

### Interaction with other applications

Composer receives information via XIMPLE and translates it to information to be sent to the configured renderers based on the Presentation file (\*.im2). Composer can receive the XIMPLE from Protran or any other source.

Composer sends the information to different Renderers if the Renderers are configured to run in System Manager and if they are configured to receive information in the Presentation file.

### Shortcuts available

If Composer is started by System Manager, it can be triggered to restart by pressing “r” + Enter in the console window.

## DirectX Renderer

C:\Users\wes\AppData\Local\Microsoft\Windows\INetCache\Content.Word\directxrenderer.emfThe DirectX Renderer application is part of Infomedia. It handles the rendering of information received from the Composer on the TFT screen. The DirectX Renderer can also be configured to render information from the Composer to multiple TFT screens.

In order to be able to render the information to the TFT screen, the system must be configured in the following files:

* **DirectXRenderer.xml** – Configure the different screens used by the DirectX Renderer.
* **Presentation file (\*.im2)** – The Physical screens, Virtual Displays, the resolution of the Layouts must all be configured in order to correctly render the information.

### Requirements to run DirectX Renderer

The DirectX Renderer must be configured to be started and handled by the System Manager. The DirectX Renderer must have a separate folder for Configuration under the “Config” folder and a separate folder for the binaries under “Progs” folder in the TFT system folder structure (refer to TD\_TFT2.0System document for details of the folder structure).

To use DirectX Renderer on a system other than InfoVision and Inform TFTs, DirectX 9c must be installed. If necessary, please install it from the following URL:

R:\Softwareserver\_Release\SW\02\_imotion\02\_TFT\00\_Basic\_System (TFT 2.0)\Requirements for Basicsystem TFT\Directx und DotNet

### Interaction with other applications

The DirectX Renderer receives the information to be rendered on the TFT screens from the Composer via Medi. The DirectX Renderer informs the Composer on some specific cases, for example, when a video has reached its end, via Medi.

### Shortcuts available

#### Alt+Enter

The DirectX Renderer can be configured to show in the following window modes (the details are available in chapter 4.2.1):

* Windowed
* FullScreenWindowed
* FullScreenExclusive

Use Alt+Enter to toggle between the different window modes.

If the TFT screen cannot be shown in FullScreenExclusive, a message is shown.

#### F1

Press “F1” to toggle displaying the following information about the rendered screen:

* Screen adapter
* Screen resolution
* Screen coordinates
* The window mode
* The current frame rate in FPS

## Audio Renderer

C:\Users\wes\AppData\Local\Microsoft\Windows\INetCache\Content.Word\audiorenderer.emfThe Audio Renderer application is part of Infomedia. It handles the rendering of information received from the Composer onto an audio device like a speaker. The Audio Renderer can play MP3 audio files and uses Acapela to output text-to-speech (TTS). The Presentation file (\*.im2) can be configured to select the audio files to be used, the text-to-speech voice, the amount of time to Pause between different audios etc.

In order to be able to render the information to an audio device, the system must be configured in the following files.

* **AudioRenderer.xml** – Configure the different IO ports to be used to control the volume and the Speaker operation (turn on or off). Configuration for the text-to-speech operation.
* **HardwareManager.xml** - Configure the IO pins on the hardware to be used for the volume and Speaker operation.
* **Presentation file (\*.im2)** – The Physical screen, Virtual Display and the Layouts must all be configured in order to correctly render the information.

### Requirements to run Audio Renderer

The Audio Renderer must be configured to be started and handled by the System Manager. The Audio Renderer must have a separate folder for Configuration under the “Config” folder and a separate folder for the binaries under “Progs” folder in the TFT system folder structure (refer to TD\_TFT2.0System document for details of the folder structure).

If the Audio Renderer is configured to use Acapela for text-to-speech, then the binaries for Acapela must be placed in the exact location as configured in the AudioRenderer.xml.

### Interaction with other applications

The Audio Renderer receives the information to be rendered on the audio device like a speaker from the Composer via Medi.

### Shortcuts available

If Audio Renderer is started by System Manager, it can be triggered to restart by pressing “r” + Enter in the console window.

## AHDLC Renderer

C:\Users\wes\AppData\Local\Microsoft\Windows\INetCache\Content.Word\ahdlcrenderer.emfThe AHDLC Renderer application is part of Infomedia. It handles the rendering of information received from the Composer to different Gorba LED signs. The AHDLC Renderer can render information to multiple LED signs at once.

In order to be able to render the information to different LED signs, the system must be configured in the following files.

* **AhdlcRenderer.xml** – Configure the serial port, the connected LED signs with the correct addresses, the type of sign and their resolutions.
* **Presentation file (\*.im2)** – Each LED sign requires a Physical Screen, Virtual Display and its associated configuration. The resolution of the LED signs here must match the resolution configured in the AhdlcRenderer.xml.

### Requirements to run AHDLC Renderer

The AHDLC Renderer must be configured to be started and handled by the System Manager. The AHDLC Renderer must have a separate folder for Configuration under the “Config” folder and a separate folder for the binaries under “Progs” folder in the TFT system folder structure (refer to TD\_TFT2.0System document for details of the folder structure).

### Interaction with other applications

The AHDLC Renderer receives the information to be rendered on the LED signs from the Composer via Medi.

### Shortcuts available

If AHDLC Renderer is started by System Manager, it can be triggered to restart by pressing “r” + Enter in the console window.

# Configuration

## Composer

The Composer requires configuration in order to operate correctly with the selected hardware. The file for configuration is **“Composer.xml”**. Composer also requires “**NLog.config**” for logging and **“medi.config”** for the Medi configuration.

### @XmlDoc(xsd=..\..\..\Common\Configuration\Source\Infomedia\Composer\Composer.xsd;xml=..\Source\Core\Composer.xml)

## DirectX Renderer

The DirectX Renderer requires configuration in order to operate correctly with the selected hardware. The file for configuration is **“DirectXRenderer.xml”**. DirectX Renderer also requires “**NLog.config**” for logging and **“medi.config”** for the Medi configuration.

### @XmlDoc(xsd=..\..\..\Common\Configuration\Source\Infomedia\DirectXRenderer\DirectXRenderer.xsd;xml=..\Source\DirectXRenderer\DirectXRenderer.xml)

## Audio Renderer

The Audio Renderer requires configuration in order to operate correctly with the selected hardware. The file for configuration is **“AudioRenderer.xml”**. Audio Renderer also requires “**NLog.config**” for logging and **“medi.config”** for the Medi configuration.

### @XmlDoc(xsd=..\..\..\Common\Configuration\Source\Infomedia\AudioRenderer\AudioRenderer.xsd;xml=..\Source\AudioRenderer\AudioRenderer.xml)

## AHDLC Renderer

The AHDLC Renderer requires configuration in order to operate correctly with the selected hardware. The file for configuration is **“AhdlcRenderer.xml”**. Ahdlc Renderer also requires “**NLog.config**” for logging and **“medi.config”** for the Medi configuration.

### @XmlDoc(xsd=..\..\..\Common\Configuration\Source\Infomedia\AhdlcRenderer\AhdlcRenderer.xsd;xml=..\Source\AhdlcRenderer\AhdlcRenderer.xml)

# Example usages

## DirectX Renderer example usages

### Define visible region on the screen example usage

In the DirectXRenderer.xml, the user can define a screen and only make a certain region of it visible. In such a configuration, the image, text etc rendered by the DirectX Renderer is shown only in the visible region. The X and Y allow the user to specify the offset of the visible region. Below is an example for such a configuration.

<Screen Adapter="0" Width="1920" Height="1080">

  <VisibleRegion X="0" Y="0" Width="1920" Height="630"/>

  <FallbackImage>..\..\Presentation\Images\Fallback.jpg</FallbackImage>

</Screen>

The example shown above is for 29’ HD display in which only the **Height** attribute is modified to value 630 in the **VisibleRegion**.

## Audio Renderer example usages

### Speaker setup example usages

The Audio Channel configuration allows the user to configure the Speaker port for a specific Hardware output. The configuration between the Hardware Manager and the Audio Renderer is given in an example in the TD\_HardwareManager.pdf document.

The second part to setup the Speaker to be switched on at a specific time is to be defined in both the **AudioRenderer.xml** and the **\*.im2** file used by the Composer application.

For each Physical Screen of type “Audio” configured in the \*.im2, an “Audio Channel” must be configured in the AudioRenderer.xml.   
**The Audio Channel and the Physical Screen are mapped to each other by the “Id”.**

Below is an example of \*.im2 with three Physical Screens of type Audio defined.

  <PhysicalScreens>

    <PhysicalScreen Name="AudioScreen1" Type="Audio" Id="1" Width="0" Height="0" />

    <PhysicalScreen Name="AudioScreen2" Type="Audio" Id="2" Width="0" Height="0" />

<PhysicalScreen Name="AudioScreen3" Type="Audio" Id="3" Width="0" Height="0" />

  </PhysicalScreens>

Below is an example of the AudioRenderer.xml with three Audio Channels configured to work with the above \*.im2.

  <IO>

    <VolumePort Name="SystemVolume" />

  </IO>

  <AudioChannels>

    <AudioChannel Id="1">

      <SpeakerPort Name="SpeakerOne" />

    </AudioChannel>

    <AudioChannel Id="2">

      <SpeakerPort Name="SpeakerTwo" />

    </AudioChannel>

    <AudioChannel Id="3">

      <SpeakerPort Name="SpeakerOne" />

<SpeakerPort Name="SpeakerTwo" />

    </AudioChannel>

 </AudioChannels>

When a cycle for Physical Screen with Id = 1 is active/enabled, the audio configured for that cycle is played on SpeakerOne which is configured under AudioChannel with Id = 1.

When a cycle for Physical Screen with Id = 3 is active/enabled, the audio configured for that cycle is played on SpeakerOne and SpeakerTwo simultaneously which are configured under AudioChannel with Id = 3.

### Acapela example usage

If the Audio Renderer is configured to use Acapela for text-to-speech, then the binaries for Acapela must be placed in the exact location as configured in the AudioRenderer.xml as seen in the example below.

  <TextToSpeech>

    <API>Acapela</API>

    <HintPath>..\..\Progs\Acapela</HintPath>

  </TextToSpeech>

As seen in the example above, the <API> must be configured as “Acapela” and the <HintPath> must be the location where the Acapela binaries are placed on the TFT system. In this example, the binaries are placed within the Progs directory in the directory called Acapela.

## AHDLC Renderer example usages

### Signs example usage

Each sign that is configured in the AHDLC Renderer must have an associated Physical Screen of type LED configured in the \*.im2 file. The “Address” attribute of the Sign is mapped to the “Id” of the Physical Screen.

Below is an example of \*.im2 with three Physical Screens of type Audio defined.

  <PhysicalScreens>

    <PhysicalScreen Name="FrontLED" Type="LED" Id="1" Width="98" Height="16" />

    <PhysicalScreen Name="SideLED" Type="LED" Id="2" Width="98" Height="16" />

    <PhysicalScreen Name="ColorLED" Type="LED" Id="12" Width="34" Height="20" />

  </PhysicalScreens>

Below is an example of the AhdlcRenderer.xml with three Signs configured to work with the above \*.im2.

    <Signs>

      <Sign Address="1" Mode="Monochrome" Width="98" Height="16" />

      <Sign Address="2" Mode="Monochrome" Width="98" Height="16" />

      <Sign Address="12" Mode="Color" Width="34" Height="20" />

    </Signs>

# Glossary

|  |  |
| --- | --- |
| AHDLC | Alpha High-Level Data Link Control; Protocol used to communicate with Gorba LED signs |
| FPS | Frames per seconds; measurement of graphics refresh speed |
| GDI | Graphics Device Interface; a core component of Microsoft Windows operating systems responsible for simple graphical rendering |
| GIOoM | Gorba I/O over Medi; Protocol used to read and write digital inputs and outputs over the Medi protocol |
| GPIO | General-purpose input/output; digital input or output available on certain hardware |
| I/O | Input or output |
| Medi | Message Dispatcher protocol used to communicate between Gorba software components |
| RTS | Request To Send; output line on a serial port |
| TTS | Text-to-speech; technology to convert normal language text into speech audio |
| VLC | VideoLAN VLC media player |
| Ximple | Protocol used over Medi for communication between Protran and Composer |