

|  |  |
| --- | --- |
|  | **imotion** |
|  |  |
|  | TFT 2.0 |
|  |  |
|  | TFT 2.0 Topbox Setup |
|  |  |
|  | Technical Description |
|  |  |
|  | Version: 0.1 |
|  | State: Draft |
|  | Classification: Internal use only |
|  | Author: RAN |
|  | Creation date: 2013-07-18 |
|  | Repository: $/Gorba/Main/Motion/Protran/Documents/TD\_ProtranDocumentation/TD\_TFT 2.0 Topbox Setup.docx |
|  | Gorba AG  Sandackerstrasse  9245 Oberbüren  Switzerland |

**Table of contents**

[1 Introduction 4](#_Toc362260641)

[1.1 Scope 4](#_Toc362260642)

[1.2 Intended Audience 4](#_Toc362260643)

[2 Folder structure on TFT 2.0 topbox 4](#_Toc362260644)

[2.1 Config folder 4](#_Toc362260645)

[2.2 Data folder 4](#_Toc362260646)

[2.3 Logs folder 4](#_Toc362260647)

[2.4 Presentation folder 5](#_Toc362260648)

[2.5 Progs folder 5](#_Toc362260649)

[3 Component Interactions with SystemManager on the topbox 5](#_Toc362260650)

[3.1 Powerup and launching of applications 5](#_Toc362260651)

[3.2 Monitoring of different applications 5](#_Toc362260652)

[3.3 Closing of applications 5](#_Toc362260653)

[3.4 Typical interaction steps between Protran and SM 5](#_Toc362260654)

[3.5 Typical interaction steps between Composer and SM 6](#_Toc362260655)

[3.6 Typical interaction steps between Renderer and SM 6](#_Toc362260656)

**Modification management**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Version** | **Date** | **Name** | **Dept.** | **Modifications** | **State** |
| 0.1 | 2013-07-19 | RAN | SW | Initial version | draft |

**Review**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date** | **Name** | **Dept.** | **Remarks** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Release**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date** | **Name** | **Dept.** | **Remarks** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# Introduction

## Scope

This document is a technical description of TFT 2.0 topbox setup. It gives all information to Gorba internal people to be able to work with setup the TFT 2.0 topbox in order to be able to use it.

This document is not intended to be a user manual.

This document also gives the details of the interactions between the different software components in the topbox. It does not describe the configuration details of the software components. Please refer to the technical document of the individual software component for detailed description of the configuration.

## Intended Audience

This document is written to be understood by Gorba staff familiar with Gorba products. Minimal technical skills are required.

# Folder structure on TFT 2.0 topbox

The folder structure for TFT 2.0 is as follows on the topbox.

* Config
* Data
* Logs
* Presentation
* Progs

## Config folder

The configuration files for all software components are placed in the “Config” folder. The Config folder has sub-folders for each software component as given below,

* Composer
* Protran
* Renderer
* SystemManager

The configuration files for the specific software components in placed within its respective folder.

## Data folder

The folder “Data” holds all the data files by the software components. This folder is used only by the software components and not touched by the user. It also contains sub-folders depending on the software components storing its respective data files for example, Protran, SystemManager are sub-folders under Data folder. An example of the data file from Protran stored in the data folder is “Persistence.xml”.

## Logs folder

The log files produced by each software component are stored in the Logs folder with the name of the log file being the name of the software component. For example, SystemManager produces the log file with name SystemManager<date>.log.

## Presentation folder

The presentation folder contains the files related to the layout and the file containing the details about the presentation, for example main.im2. This folder is edited by the user.

## Progs folder

The software component applications are placed in the Progs folder. The Progs folder has sub-folders for each software component as given below,

* Composer
* Protran
* Renderer
* Startup
* SystemManager

The application files for the specific software components in placed within its respective folder.

# Component Interactions with SystemManager on the topbox

The SystemManager(SM) configuration file, SystemManager.xml contains the configuration settings to allow SM to control and communicate with all the software components on the topbox.

## Powerup and launching of applications

Upon powerup, SM is started. An initial splash screen is shown indicating the status to SM and all the applications/components to be launched by SM. SM then launches all the software components specified in SystemManger.xml. Each component launched by SM communicated its status such as Launching, Running etc.

## Monitoring of different applications

Each application/component launched by SM is monitored based on the configuration for the specified application in SystemManager.xml. Some of the parameters that can be monitored are the CPU and Ram usage for the specific application. Based the limits set for the CPU and RAM usage and the subsequent action for a specific application, SM closes or restarts the application.

## Closing of applications

Applications controlled by SM, are also closed from SM. If an application is to be closed by SM, then it informs the application to close. The application then closes and informs SM that it is closed. If an application wants to close, for example, if the user chooses to close the application by pressing “q” to exit, then the application lets SM know that it must be closed. SM then registers that information and then informs the application to close.

## Typical interaction steps between Protran and SM

1. SM launches Protran with Protran.exe.
2. Protran then starts and based on Protran.xml starts the required protocols.
3. Once, the protocols are started, Protran informs SM that it is in the “running” state.
4. When Protran is to be exited by the user or by Protran itself due to certain conditions, then it informs SM that it must exit and specifies the reason for exit. SM then informs Protran to close. Upon receiving the close from SM, Protran exits gracefully.
5. SM also observes the ibis reading loop. If the reading on the ibis port fails, then SM is informed and the watchdog for Protran is not reset. Hence, SM will wait for the watchdog timeout and then relaunch Protran.

## Typical interaction steps between Composer and SM

1. SM launches Composer with Composer.exe.
2. Composer then starts by either using the default presentation file main.im2 or the one specified by the user by arguments in the console window.
3. Once, the presentation is started, Composer informs SM that it is in the “running” state.
4. When Composer is to be exited by the user or by Composer itself due to certain conditions, then it informs SM that it must exit and specifies the reason for exit. SM then informs Composer to close. Upon receiving the close from SM, Composer exits gracefully.

## Typical interaction steps between Renderer and SM

1. SM launches Renderer with DirectXRenderer.exe.
2. DirectX Renderer then starts and informs SM that it is in the “running” state.
3. When DirectX Renderer is to be exited by the user or by DirectX Renderer itself due to certain conditions, then it informs SM that it must exit and specifies the reason for exit. SM then informs DirectX Renderer to close. Upon receiving the close from SM, DirectX Renderer exits gracefully.
4. SM also observes the rendering loop. If the rendering fails, then SM is informed and the watchdog for DirectX Renderer is not reset. Hence, SM will wait for the watchdog timeout and then relaunch DirectX Renderer.