Inhoud

[1 Nota’s en Eigendommen 1](#_Toc41598502)

[2 Media 1](#_Toc41598503)

[2.1 Data model 1](#_Toc41598504)

[2.2 Media Database Application 2](#_Toc41598505)

[2.2.1 goedegep.media.mediadb.albuminfo 4](#_Toc41598506)

[3 Rolodex 6](#_Toc41598507)

[4 Unit Converter 6](#_Toc41598508)

[5 Vacations 6](#_Toc41598509)

[6 PCTools 6](#_Toc41598510)

[7 External Projects 6](#_Toc41598511)

[7.1 Gluon Maps 6](#_Toc41598512)

# TEMP Dialogs

## What do I need?

**General layout:**

* Title in title bar (parameter)
* Icon in title bar (customization)
* Icon to indicate type of dialog:
  + Error dialog: X
  + Option dialog :?

**Dialog types:**

* Error dialog (information only)  
  Use case: report an error to the user.
  + Icon to indicate type of dialog: X
  + text to explain the error
  + OK button
* Warning dialog (information only)  
  Use case: warn the user about something.
  + Icon to indicate type of dialog: !
  + text with the warning
  + OK button
* Option dialog  
  Use case: user has to select one of several options, before the application can continue.
  + Icon: ?
  + text to explain the option selection
  + radio buttons to select option
  + OK button
  + Do I need a cancel button? (not for now)
* FileReference dialog  
  Use case: create a new FileReference.
  + No icon
  + text to explain the FileReference
  + fields to select a file
  + fields to specify a title
  + OK button – FileReference will be created
  + Cancel button – No action.

## Solution

* Stay as close as possible to the JavaFx dialog API.
* Error dialog -> Alert, type is ERROR
* Warning dialog -> Alert, type is WARNING
* Option dialog -> ChoiceDialog (combo box instead of radio buttons).
* FileReference dialog: to be created.

# Media

## Media Database Application

Hi

AlbumsTable

albumsTable

MediaDbWindow

TracksTable

**goedegep.media.mediadb.app.guifx**

**goedegep.media.model.mediadb**

**goedegep.appgenfx**

tracksTable

**goedegep.util**

AlbumDiscLocationInfo

**goedegep.media.mediadb.app**

**goedegep.media.mediadb.albuminfo**

### goedegep.media.mediadb.albuminfo

This package has all the knowledge of the structure and file name conventions of the AlbumInfo files. Information read from the AlbumInfo files is stored in the MediaDb data model, and a MediaDb data model can be written to AlbumInfo files.

AlbumInfoHandler

albumInfoHandler

AlbumInfoFilesReader

AlbumInfoErrorInfo

**goedegep.media.mediadb.albuminfo**

**goedegep.media.model.mediadb**

**goedegep.util**

AlbumInfoError

errorCode

errors

* **AlbumInfoFilesReader**  
  This class reads 'AlbumInfo' files and stores the information in a **MediaDb** structure. It also writes the information from a **MediaDb** structure to 'AlbumInfo' files.  
  This class has the knowledge of the filename conventions. For reading it scans the folder with 'AlbumInfo' files. For each file it calls an **AlbumInfoHandler** to parse the file.
* **AlbumInfoHandler**This class reads 'AlbumInfo' XML files and stores the information in a **MediaDb** model.  
  The schema for these files is defined in the xsd file "D:\Database\Muziek\Albums\AlbumInfo.xsd".  
  The parser can generate errors of the following types: **AlbumInfoErrorInfo**, **goedegep.util.ParseException**.

# Rolodex

This component has sub components, so it uses a component launcher, class RolodexMenuWindow.

# Unit Converter

This component is a single application, where the main class is UnitConverterWindow.

# Vacations

This component is a single application, where the main class is VacationsWindow.

# PCTools

This component has sub components, so it uses a component launcher, class PCToolsMenuWindow.  
The sub components are:

* FileFinder
* FilesControlled

# External Projects

## Gluon Maps

This project provides a map view.

MapPoint

MapView

javafx.scene.layout.Region

BaseMap

baseMap

MapLayer

layers

javafx.scene.Parent

javafx.scene.Group

**JavaFx**

**com.gluonhq.maps**

* **com.gluonhq.maps**
  + **MapPoint**  
    This class provides a point on a map, which are just the latitude/longitude coordinates.
  + **MapView**This is the top UI element of the map component.  
    This class is mainly a wrapper around a **BaseMap**.
  + **MapLayer**A MapLayer can be added on top of a BaseMap (which provides the map tiles).  
    This class only provides the framework for map layer. Any specific map layer shall extend this class and override the methods: initialize() and layoutLayer().
* **com.gluonhq.impl.maps**
  + **BaseMap**  
    The BaseMap provides the underlying map tiles. On top of this, additional layers can be rendered.