

Rodoc

Generated by Doxygen 1.9.7



|                                       |           |
|---------------------------------------|-----------|
| <b>1 Rodoc</b>                        | <b>1</b>  |
| <b>2 icons</b>                        | <b>3</b>  |
| <b>3 Events</b>                       | <b>5</b>  |
| 3.1 Event Templates . . . . .         | 5         |
| 3.2 Predefined Templates . . . . .    | 5         |
| <b>4 Files, Notes &amp; Media</b>     | <b>7</b>  |
| 4.1 Virtual Directories . . . . .     | 7         |
| 4.2 Adding Files . . . . .            | 7         |
| <b>5 Welcome to rodoc</b>             | <b>9</b>  |
| 5.1 Project . . . . .                 | 9         |
| 5.2 Persons . . . . .                 | 9         |
| 5.3 Relations . . . . .               | 9         |
| 5.4 Events . . . . .                  | 9         |
| 5.5 Files . . . . .                   | 9         |
| <b>6 Layout</b>                       | <b>11</b> |
| 6.1 Menu . . . . .                    | 11        |
| 6.2 Panels . . . . .                  | 11        |
| 6.2.1 Left Panel . . . . .            | 11        |
| 6.2.2 Right Panel . . . . .           | 11        |
| <b>7 Person</b>                       | <b>13</b> |
| 7.1 Dates . . . . .                   | 13        |
| 7.2 Tags . . . . .                    | 13        |
| <b>8 Relations</b>                    | <b>15</b> |
| 8.1 Relation Templates . . . . .      | 15        |
| 8.1.1 Generation Difference . . . . . | 15        |
| 8.1.2 Trait . . . . .                 | 16        |
| 8.1.3 Predefined Templates . . . . .  | 16        |
| 8.2 Promotion . . . . .               | 16        |
| 8.3 Suggestions . . . . .             | 16        |
| <b>9 Namespace Index</b>              | <b>17</b> |
| 9.1 Namespace List . . . . .          | 17        |
| <b>10 Hierarchical Index</b>          | <b>19</b> |
| 10.1 Class Hierarchy . . . . .        | 19        |
| <b>11 Class Index</b>                 | <b>21</b> |
| 11.1 Class List . . . . .             | 21        |

|                                   |           |
|-----------------------------------|-----------|
| <b>12 File Index</b>              | <b>23</b> |
| 12.1 File List                    | 23        |
| <b>13 Namespace Documentation</b> | <b>25</b> |
| 13.1 addfile Namespace Reference  | 25        |
| 13.1.1 Detailed Description       | 25        |
| 13.1.2 Variable Documentation     | 25        |
| 13.1.2.1 NONE                     | 25        |
| 13.2 button Namespace Reference   | 25        |
| 13.2.1 Detailed Description       | 26        |
| 13.2.2 Variable Documentation     | 26        |
| 13.2.2.1 ADD_NEW                  | 26        |
| 13.2.2.2 CANCEL                   | 26        |
| 13.2.2.3 REMOVE                   | 26        |
| 13.3 error Namespace Reference    | 26        |
| 13.3.1 Detailed Description       | 27        |
| 13.3.2 Variable Documentation     | 27        |
| 13.3.2.1 CORRUPTED_DATABASE       | 27        |
| 13.3.2.2 EMPTY_BOXES              | 27        |
| 13.3.2.3 EXCEPTION_STR            | 27        |
| 13.3.2.4 FORBIDDEN_CHARS          | 27        |
| 13.3.2.5 NO_CHOSEN_FILE           | 28        |
| 13.3.2.6 NO_ROLE                  | 28        |
| 13.3.2.7 UNKNOWN_EXCEP_STR        | 28        |
| 13.4 events Namespace Reference   | 28        |
| 13.4.1 Detailed Description       | 29        |
| 13.4.2 Variable Documentation     | 29        |
| 13.4.2.1 BEST_MAN                 | 29        |
| 13.4.2.2 BRIDE                    | 29        |
| 13.4.2.3 DEAD_PERSON              | 29        |
| 13.4.2.4 DESC_DIVORCE             | 29        |
| 13.4.2.5 DESC_FUNERAL             | 30        |
| 13.4.2.6 DESC_GRADUATION          | 30        |
| 13.4.2.7 DESC_WEDDING             | 30        |
| 13.4.2.8 DIVORCE                  | 30        |
| 13.4.2.9 FORMER_HUSBAND           | 30        |
| 13.4.2.10 FORMER_WIFE             | 30        |
| 13.4.2.11 FUNERAL                 | 30        |
| 13.4.2.12 GRADUATE                | 31        |
| 13.4.2.13 GRADUATION              | 31        |
| 13.4.2.14 GROOM                   | 31        |
| 13.4.2.15 MAID_OF_HONOR           | 31        |

|                                    |    |
|------------------------------------|----|
| 13.4.2.16 NO_ROLE                  | 31 |
| 13.4.2.17 WEDDING                  | 31 |
| 13.5 gender Namespace Reference    | 31 |
| 13.5.1 Detailed Description        | 32 |
| 13.5.2 Function Documentation      | 32 |
| 13.5.2.1 str()                     | 32 |
| 13.5.3 Variable Documentation      | 32 |
| 13.5.3.1 AllGenders                | 32 |
| 13.6 html Namespace Reference      | 32 |
| 13.6.1 Detailed Description        | 33 |
| 13.6.2 Variable Documentation      | 34 |
| 13.6.2.1 ALL_PERSONS               | 34 |
| 13.6.2.2 BIRTH_DATE                | 34 |
| 13.6.2.3 BIRTH_PLACE               | 34 |
| 13.6.2.4 CLOSE_FAMILY              | 34 |
| 13.6.2.5 CLOSE_FAMILY_DIAGRAM      | 34 |
| 13.6.2.6 CUSTOM_CSS                | 34 |
| 13.6.2.7 CUSTOM_TAGS               | 34 |
| 13.6.2.8 DATE                      | 35 |
| 13.6.2.9 DEATH_DATE                | 35 |
| 13.6.2.10 DEATH_PLACE              | 35 |
| 13.6.2.11 DESCRIPTION              | 35 |
| 13.6.2.12 EVENTS                   | 35 |
| 13.6.2.13 FAMILY_TREE              | 35 |
| 13.6.2.14 FAMILY_TREE_DIAGRAM      | 35 |
| 13.6.2.15 GENDER                   | 35 |
| 13.6.2.16 GENERAL_INFO             | 36 |
| 13.6.2.17 NOTE                     | 36 |
| 13.6.2.18 PERSONS                  | 36 |
| 13.6.2.19 PLACE                    | 36 |
| 13.6.2.20 PREFIX_CLOSE             | 36 |
| 13.6.2.21 PREFIX_FAMILY_TREE       | 36 |
| 13.6.2.22 RELATIONS                | 36 |
| 13.6.2.23 RESOURCES                | 36 |
| 13.7 jsonlabel Namespace Reference | 37 |
| 13.7.1 Detailed Description        | 40 |
| 13.7.2 Variable Documentation      | 40 |
| 13.7.2.1 AFTER_TITLE               | 40 |
| 13.7.2.2 APP_SETTINGS              | 40 |
| 13.7.2.3 BIRTH                     | 40 |
| 13.7.2.4 BLUE                      | 40 |
| 13.7.2.5 BORDER_RADIUS             | 41 |

|  |    |
|--|----|
| 13.7.2.6 CONTAINS_DATE . . . . .       | 41 |
| 13.7.2.7 CONTAINS_PLACE . . . . .      | 41 |
| 13.7.2.8 DATE . . . . .                | 41 |
| 13.7.2.9 DEATH . . . . .               | 41 |
| 13.7.2.10 DESCRIPTION . . . . .        | 41 |
| 13.7.2.11 DIR . . . . .                | 41 |
| 13.7.2.12 DRIVES . . . . .             | 41 |
| 13.7.2.13 EVE_TEMPL . . . . .          | 42 |
| 13.7.2.14 EVENTS . . . . .             | 42 |
| 13.7.2.15 EXPORT . . . . .             | 42 |
| 13.7.2.16 FATHER . . . . .             | 42 |
| 13.7.2.17 FILES . . . . .              | 42 |
| 13.7.2.18 FIRST_NAME . . . . .         | 42 |
| 13.7.2.19 FIRST_PERSON . . . . .       | 42 |
| 13.7.2.20 FONT_FAMILY . . . . .        | 42 |
| 13.7.2.21 FONT_SIZE . . . . .          | 43 |
| 13.7.2.22 FRONT_TITLE . . . . .        | 43 |
| 13.7.2.23 GEN_DIFF . . . . .           | 43 |
| 13.7.2.24 GEN_SIZE_DOWN . . . . .      | 43 |
| 13.7.2.25 GEN_SIZE_UP . . . . .        | 43 |
| 13.7.2.26 GENDER . . . . .             | 43 |
| 13.7.2.27 GLOBAL_MAIN_PERSON . . . . . | 43 |
| 13.7.2.28 GREEN . . . . .              | 43 |
| 13.7.2.29 HELP . . . . .               | 44 |
| 13.7.2.30 HIGHLIGHTED_COLOR . . . . .  | 44 |
| 13.7.2.31 ID . . . . .                 | 44 |
| 13.7.2.32 LAST_DATE . . . . .          | 44 |
| 13.7.2.33 LINE_COLOR . . . . .         | 44 |
| 13.7.2.34 LINE_WIDTH . . . . .         | 44 |
| 13.7.2.35 LIVES . . . . .              | 44 |
| 13.7.2.36 MAIDEN_NAME . . . . .        | 44 |
| 13.7.2.37 MEDIA . . . . .              | 45 |
| 13.7.2.38 MORE_PEOPLE . . . . .        | 45 |
| 13.7.2.39 MOTHER . . . . .             | 45 |
| 13.7.2.40 MOVEABLE_TABS . . . . .      | 45 |
| 13.7.2.41 NAME . . . . .               | 45 |
| 13.7.2.42 NOTES . . . . .              | 45 |
| 13.7.2.43 PARTNER . . . . .            | 45 |
| 13.7.2.44 PERSON . . . . .             | 45 |
| 13.7.2.45 PERSONS . . . . .            | 46 |
| 13.7.2.46 PLACE . . . . .              | 46 |
| 13.7.2.47 PROBANT_COLOR . . . . .      | 46 |

---

|   |    |
|---|----|
| 13.7.2.48 PROMOTED_COLOR . . . . .        | 46 |
| 13.7.2.49 RED . . . . .                   | 46 |
| 13.7.2.50 REL_TEMPL . . . . .             | 46 |
| 13.7.2.51 RELATIONS . . . . .             | 46 |
| 13.7.2.52 ROLE . . . . .                  | 46 |
| 13.7.2.53 ROOT . . . . .                  | 47 |
| 13.7.2.54 SECOND_NAME . . . . .           | 47 |
| 13.7.2.55 SECOND_PERSON . . . . .         | 47 |
| 13.7.2.56 SHOW_EVENT_TAB . . . . .        | 47 |
| 13.7.2.57 SHOW_FILE_TAB . . . . .         | 47 |
| 13.7.2.58 SHOW_GENERAL_TAB . . . . .      | 47 |
| 13.7.2.59 SHOW_MEDIA_TAB . . . . .        | 47 |
| 13.7.2.60 SHOW_NOTE_TAB . . . . .         | 47 |
| 13.7.2.61 SHOW_RELATION_TAB . . . . .     | 48 |
| 13.7.2.62 SPECIAL_ROLES . . . . .         | 48 |
| 13.7.2.63 SPLITTER_POSITION_ONE . . . . . | 48 |
| 13.7.2.64 SPLITTER_POSITION_TWO . . . . . | 48 |
| 13.7.2.65 STANDARD_COLOR . . . . .        | 48 |
| 13.7.2.66 SURNAME . . . . .               | 48 |
| 13.7.2.67 TAB_POSITION . . . . .          | 48 |
| 13.7.2.68 TAG . . . . .                   | 48 |
| 13.7.2.69 TAGS . . . . .                  | 49 |
| 13.7.2.70 TEMPLATE . . . . .              | 49 |
| 13.7.2.71 TEMPLATES . . . . .             | 49 |
| 13.7.2.72 TEXT . . . . .                  | 49 |
| 13.7.2.73 TEXT_COLOR . . . . .            | 49 |
| 13.7.2.74 TITLE . . . . .                 | 49 |
| 13.7.2.75 TRAIT . . . . .                 | 49 |
| 13.7.2.76 TYPE . . . . .                  | 49 |
| 13.7.2.77 VALUE . . . . .                 | 50 |
| 13.7.2.78 VIRTUAL_DRIVE . . . . .         | 50 |
| 13.7.2.79 X_SIZE . . . . .                | 50 |
| 13.7.2.80 Y_SIZE . . . . .                | 50 |
| 13.8 parser Namespace Reference . . . . . | 50 |
| 13.8.1 Detailed Description . . . . .     | 51 |
| 13.8.2 Variable Documentation . . . . .   | 51 |
| 13.8.2.1 ERROR_LOG . . . . .              | 51 |
| 13.8.2.2 FILES_DIR . . . . .              | 51 |
| 13.8.2.3 JSON_CONFIG . . . . .            | 51 |
| 13.8.2.4 JSON_EVENTS . . . . .            | 51 |
| 13.8.2.5 JSON_FILES . . . . .             | 51 |
| 13.8.2.6 JSON_MEDIA . . . . .             | 51 |

|   |    |
|---|----|
| 13.8.2.7 JSON_NOTES . . . . .                 | 52 |
| 13.8.2.8 JSON_PERSONS . . . . .               | 52 |
| 13.8.2.9 JSON_RELATIONS . . . . .             | 52 |
| 13.8.2.10 MEDIA_DIR . . . . .                 | 52 |
| 13.8.2.11 NOTES_DIR . . . . .                 | 52 |
| 13.9 person Namespace Reference . . . . .     | 52 |
| 13.9.1 Detailed Description . . . . .         | 53 |
| 13.9.2 Variable Documentation . . . . .       | 53 |
| 13.9.2.1 DATE_FROM . . . . .                  | 53 |
| 13.9.2.2 DATE_TO . . . . .                    | 53 |
| 13.9.2.3 NAME_STR . . . . .                   | 53 |
| 13.9.2.4 ROOT . . . . .                       | 53 |
| 13.9.2.5 SURNAME_STR . . . . .                | 53 |
| 13.9.2.6 UNKNOWN_STR . . . . .                | 53 |
| 13.10 relation Namespace Reference . . . . .  | 53 |
| 13.10.1 Detailed Description . . . . .        | 55 |
| 13.10.2 Variable Documentation . . . . .      | 55 |
| 13.10.2.1 ADOPTED_CHILD . . . . .             | 55 |
| 13.10.2.2 ADOPTIVE_FATHER . . . . .           | 55 |
| 13.10.2.3 ADOPTIVE_FATHERHOOD . . . . .       | 55 |
| 13.10.2.4 ADOPTIVE_MOTHER . . . . .           | 55 |
| 13.10.2.5 ADOPTIVE_MOTHERHOOD . . . . .       | 55 |
| 13.10.2.6 BROTHER . . . . .                   | 56 |
| 13.10.2.7 BROTHERHHOD . . . . .               | 56 |
| 13.10.2.8 CHILD . . . . .                     | 56 |
| 13.10.2.9 DESC_ADOPTIVE_FATHERHOOD . . . . .  | 56 |
| 13.10.2.10 DESC_ADOPTIVE_MOTHERHOOD . . . . . | 56 |
| 13.10.2.11 DESC_BROTHERHOOD . . . . .         | 56 |
| 13.10.2.12 DESC_SIBLINGS . . . . .            | 56 |
| 13.10.2.13 DESC_SISTERHOOD . . . . .          | 57 |
| 13.10.2.14 FATHER . . . . .                   | 57 |
| 13.10.2.15 FATHERHOOD . . . . .               | 57 |
| 13.10.2.16 FATHERHOOD_DESC . . . . .          | 57 |
| 13.10.2.17 HUSBAND . . . . .                  | 57 |
| 13.10.2.18 MARRIAGE . . . . .                 | 57 |
| 13.10.2.19 MARRIAGE_DESC . . . . .            | 57 |
| 13.10.2.20 MOTHER . . . . .                   | 58 |
| 13.10.2.21 MOTHERHOOD . . . . .               | 58 |
| 13.10.2.22 MOTHERHOOD_DESC . . . . .          | 58 |
| 13.10.2.23 PARTNER . . . . .                  | 58 |
| 13.10.2.24 PARTNERSHIP . . . . .              | 58 |
| 13.10.2.25 PARTNERSHIP_DESC . . . . .         | 58 |



|   |           |
|---|-----------|
| 13.10.2.26 SIBLINGS . . . . .                           | 58        |
| 13.10.2.27 SISTER . . . . .                             | 58        |
| 13.10.2.28 SISTERHOOD . . . . .                         | 59        |
| 13.10.2.29 WIFE . . . . .                               | 59        |
| 13.11 tagchooser Namespace Reference . . . . .          | 59        |
| 13.11.1 Detailed Description . . . . .                  | 59        |
| 13.11.2 Variable Documentation . . . . .                | 59        |
| 13.11.2.1 CUSTOM_TAG . . . . .                          | 59        |
| 13.12 trait Namespace Reference . . . . .               | 59        |
| 13.12.1 Detailed Description . . . . .                  | 60        |
| 13.12.2 Function Documentation . . . . .                | 60        |
| 13.12.2.1 str() . . . . .                               | 60        |
| 13.12.3 Variable Documentation . . . . .                | 60        |
| 13.12.3.1 AllTraits . . . . .                           | 60        |
| 13.13 Ui Namespace Reference . . . . .                  | 60        |
| 13.13.1 Detailed Description . . . . .                  | 60        |
| <b>14 Class Documentation</b>                           | <b>61</b> |
| 14.1 AddFileDialog Class Reference . . . . .            | 61        |
| 14.1.1 Detailed Description . . . . .                   | 62        |
| 14.1.2 Constructor & Destructor Documentation . . . . . | 62        |
| 14.1.2.1 AddFileDialog() . . . . .                      | 62        |
| 14.1.2.2 ~AddFileDialog() . . . . .                     | 62        |
| 14.1.3 Member Function Documentation . . . . .          | 62        |
| 14.1.3.1 copyFile . . . . .                             | 62        |
| 14.1.3.2 createOrChooseFile . . . . .                   | 63        |
| 14.1.3.3 getId() . . . . .                              | 63        |
| 14.1.3.4 openFile . . . . .                             | 63        |
| 14.1.4 Member Data Documentation . . . . .              | 63        |
| 14.1.4.1 drive_ . . . . .                               | 63        |
| 14.1.4.2 FT_ . . . . .                                  | 63        |
| 14.1.4.3 type_ . . . . .                                | 64        |
| 14.1.4.4 ui . . . . .                                   | 64        |
| 14.2 AppSettings Struct Reference . . . . .             | 64        |
| 14.2.1 Detailed Description . . . . .                   | 65        |
| 14.2.2 Member Function Documentation . . . . .          | 65        |
| 14.2.2.1 readJson() . . . . .                           | 65        |
| 14.2.2.2 setDefault() . . . . .                         | 66        |
| 14.2.3 Member Data Documentation . . . . .              | 66        |
| 14.2.3.1 fontFamily . . . . .                           | 66        |
| 14.2.3.2 fontSize . . . . .                             | 66        |
| 14.2.3.3 genSizeDown . . . . .                          | 66        |

|           |  |    |
|-----------|--|----|
| 14.2.3.4  | genSizeUp                              | 66 |
| 14.2.3.5  | highlightedColor                       | 66 |
| 14.2.3.6  | lineColor                              | 66 |
| 14.2.3.7  | lineWidth                              | 67 |
| 14.2.3.8  | probandColor                           | 67 |
| 14.2.3.9  | promoted                               | 67 |
| 14.2.3.10 | radius                                 | 67 |
| 14.2.3.11 | showEventTab                           | 67 |
| 14.2.3.12 | showFileTab                            | 67 |
| 14.2.3.13 | showGeneralTab                         | 67 |
| 14.2.3.14 | showMediaTab                           | 67 |
| 14.2.3.15 | showNoteTab                            | 68 |
| 14.2.3.16 | showRelationTab                        | 68 |
| 14.2.3.17 | sizeX                                  | 68 |
| 14.2.3.18 | sizeY                                  | 68 |
| 14.2.3.19 | splitterPositionOne                    | 68 |
| 14.2.3.20 | splitterPositionTwo                    | 68 |
| 14.2.3.21 | standard                               | 68 |
| 14.2.3.22 | tabPosition                            | 68 |
| 14.2.3.23 | textColor                              | 69 |
| 14.3      | CellGraphicsItem Class Reference       | 69 |
| 14.3.1    | Detailed Description                   | 70 |
| 14.3.2    | Constructor & Destructor Documentation | 70 |
| 14.3.2.1  | CellGraphicsItem()                     | 70 |
| 14.3.3    | Member Function Documentation          | 70 |
| 14.3.3.1  | boundingRect()                         | 70 |
| 14.3.3.2  | paint()                                | 70 |
| 14.3.3.3  | shape()                                | 71 |
| 14.3.4    | Member Data Documentation              | 71 |
| 14.3.4.1  | borderRadius_                          | 71 |
| 14.3.4.2  | pen_                                   | 71 |
| 14.3.4.3  | sizeX_                                 | 71 |
| 14.3.4.4  | sizeY_                                 | 71 |
| 14.3.4.5  | x_                                     | 71 |
| 14.3.4.6  | y_                                     | 72 |
| 14.4      | ChoosePersonDialog Class Reference     | 72 |
| 14.4.1    | Detailed Description                   | 73 |
| 14.4.2    | Constructor & Destructor Documentation | 73 |
| 14.4.2.1  | ChoosePersonDialog()                   | 73 |
| 14.4.2.2  | ~ChoosePersonDialog()                  | 73 |
| 14.4.3    | Member Function Documentation          | 73 |
| 14.4.3.1  | clearFilter                            | 73 |

---

|   |    |
|---|----|
| 14.4.3.2 filterPersons . . . . .                        | 74 |
| 14.4.3.3 savePerson . . . . .                           | 74 |
| 14.4.3.4 saveSelectedPerson . . . . .                   | 74 |
| 14.4.4 Member Data Documentation . . . . .              | 74 |
| 14.4.4.1 FT_ . . . . .                                  | 74 |
| 14.4.4.2 person_ . . . . .                              | 74 |
| 14.4.4.3 ui . . . . .                                   | 74 |
| 14.5 Color Struct Reference . . . . .                   | 75 |
| 14.5.1 Detailed Description . . . . .                   | 75 |
| 14.5.2 Constructor & Destructor Documentation . . . . . | 75 |
| 14.5.2.1 Color() . . . . .                              | 75 |
| 14.5.3 Member Function Documentation . . . . .          | 76 |
| 14.5.3.1 readJson() . . . . .                           | 76 |
| 14.5.4 Member Data Documentation . . . . .              | 76 |
| 14.5.4.1 blue . . . . .                                 | 76 |
| 14.5.4.2 green . . . . .                                | 76 |
| 14.5.4.3 red . . . . .                                  | 76 |
| 14.6 Date Class Reference . . . . .                     | 76 |
| 14.6.1 Detailed Description . . . . .                   | 77 |
| 14.6.2 Constructor & Destructor Documentation . . . . . | 77 |
| 14.6.2.1 Date() . . . . .                               | 77 |
| 14.6.3 Member Function Documentation . . . . .          | 78 |
| 14.6.3.1 getDay() . . . . .                             | 78 |
| 14.6.3.2 getMonth() . . . . .                           | 78 |
| 14.6.3.3 getYear() . . . . .                            | 78 |
| 14.6.3.4 isEmpty() . . . . .                            | 78 |
| 14.6.3.5 isExact() . . . . .                            | 78 |
| 14.6.3.6 read() . . . . .                               | 78 |
| 14.6.3.7 setDay() . . . . .                             | 79 |
| 14.6.3.8 setMonth() . . . . .                           | 79 |
| 14.6.3.9 setYear() . . . . .                            | 79 |
| 14.6.3.10 str() . . . . .                               | 79 |
| 14.6.4 Member Data Documentation . . . . .              | 80 |
| 14.6.4.1 day . . . . .                                  | 80 |
| 14.6.4.2 month . . . . .                                | 80 |
| 14.6.4.3 year . . . . .                                 | 80 |
| 14.7 DateDialog Class Reference . . . . .               | 80 |
| 14.7.1 Detailed Description . . . . .                   | 81 |
| 14.7.2 Constructor & Destructor Documentation . . . . . | 81 |
| 14.7.2.1 DateDialog() . . . . .                         | 81 |
| 14.7.2.2 ~DateDialog() . . . . .                        | 81 |
| 14.7.3 Member Function Documentation . . . . .          | 81 |

|   |    |
|---|----|
| 14.7.3.1 saveDate . . . . .                             | 81 |
| 14.7.3.2 unknownDate . . . . .                          | 82 |
| 14.7.4 Member Data Documentation . . . . .              | 82 |
| 14.7.4.1 date_ . . . . .                                | 82 |
| 14.7.4.2 FT_ . . . . .                                  | 82 |
| 14.7.4.3 ui . . . . .                                   | 82 |
| 14.8 Event Class Reference . . . . .                    | 82 |
| 14.8.1 Detailed Description . . . . .                   | 83 |
| 14.8.2 Constructor & Destructor Documentation . . . . . | 83 |
| 14.8.2.1 Event() . . . . .                              | 83 |
| 14.8.3 Member Function Documentation . . . . .          | 84 |
| 14.8.3.1 addPerson() . . . . .                          | 84 |
| 14.8.3.2 clearPersons() . . . . .                       | 84 |
| 14.8.3.3 date() . . . . .                               | 84 |
| 14.8.3.4 getDate() . . . . .                            | 84 |
| 14.8.3.5 getId() . . . . .                              | 85 |
| 14.8.3.6 getPersons() . . . . .                         | 85 |
| 14.8.3.7 getPlace() . . . . .                           | 85 |
| 14.8.3.8 getSettings() . . . . .                        | 85 |
| 14.8.3.9 getTemplate() . . . . .                        | 85 |
| 14.8.3.10 getText() . . . . .                           | 86 |
| 14.8.3.11 printHtml() . . . . .                         | 86 |
| 14.8.3.12 readJson() . . . . .                          | 86 |
| 14.8.3.13 removePerson() . . . . .                      | 86 |
| 14.8.3.14 setDate() . . . . .                           | 87 |
| 14.8.3.15 setId() . . . . .                             | 87 |
| 14.8.3.16 setPlace() . . . . .                          | 87 |
| 14.8.3.17 setTemplate() . . . . .                       | 87 |
| 14.8.3.18 setText() . . . . .                           | 88 |
| 14.8.3.19 str() . . . . .                               | 88 |
| 14.8.3.20 updateToTemplate() . . . . .                  | 88 |
| 14.8.4 Member Data Documentation . . . . .              | 88 |
| 14.8.4.1 date_ . . . . .                                | 88 |
| 14.8.4.2 id_ . . . . .                                  | 88 |
| 14.8.4.3 persons_ . . . . .                             | 89 |
| 14.8.4.4 place_ . . . . .                               | 89 |
| 14.8.4.5 settings_ . . . . .                            | 89 |
| 14.8.4.6 template_ . . . . .                            | 89 |
| 14.8.4.7 text_ . . . . .                                | 89 |
| 14.9 EventDialog Class Reference . . . . .              | 89 |
| 14.9.1 Detailed Description . . . . .                   | 90 |
| 14.9.2 Constructor & Destructor Documentation . . . . . | 91 |

|  |    |
|--|----|
| 14.9.2.1 EventDialog()                         | 91 |
| 14.9.2.2 ~EventDialog()                        | 91 |
| 14.9.3 Member Function Documentation           | 91 |
| 14.9.3.1 addEmptyRole                          | 91 |
| 14.9.3.2 choosePerson                          | 91 |
| 14.9.3.3 openCustomDate                        | 91 |
| 14.9.3.4 refreshDates                          | 91 |
| 14.9.3.5 refreshPersons                        | 91 |
| 14.9.3.6 removeEmptyRole                       | 92 |
| 14.9.3.7 removePerson                          | 92 |
| 14.9.3.8 saveEvent                             | 92 |
| 14.9.3.9 savePerson                            | 92 |
| 14.9.3.10 showEventDetails()                   | 92 |
| 14.9.3.11 showTemplate                         | 92 |
| 14.9.4 Member Data Documentation               | 93 |
| 14.9.4.1 date_                                 | 93 |
| 14.9.4.2 event_                                | 93 |
| 14.9.4.3 FT_                                   | 93 |
| 14.9.4.4 main_                                 | 93 |
| 14.9.4.5 ui                                    | 93 |
| 14.10 EventRoleItem Class Reference            | 93 |
| 14.10.1 Detailed Description                   | 94 |
| 14.10.2 Constructor & Destructor Documentation | 94 |
| 14.10.2.1 EventRoleItem()                      | 94 |
| 14.10.3 Member Function Documentation          | 94 |
| 14.10.3.1 getId()                              | 94 |
| 14.10.3.2 setPerson()                          | 94 |
| 14.10.4 Member Data Documentation              | 95 |
| 14.10.4.1 id_                                  | 95 |
| 14.11 EventTemplatesDialog Class Reference     | 95 |
| 14.11.1 Detailed Description                   | 96 |
| 14.11.2 Constructor & Destructor Documentation | 96 |
| 14.11.2.1 EventTemplatesDialog()               | 96 |
| 14.11.2.2 ~EventTemplatesDialog()              | 96 |
| 14.11.3 Member Function Documentation          | 97 |
| 14.11.3.1 addDefaults                          | 97 |
| 14.11.3.2 addNewRole                           | 97 |
| 14.11.3.3 addNewTemplate                       | 97 |
| 14.11.3.4 clear()                              | 97 |
| 14.11.3.5 refreshListWidget()                  | 97 |
| 14.11.3.6 removeRole                           | 97 |
| 14.11.3.7 removeTemplate                       | 97 |

|  |     |
|--|-----|
| 14.11.3.8 saveTemplate [1/2]                   | 97  |
| 14.11.3.9 saveTemplate() [2/2]                 | 97  |
| 14.11.3.10 setEnabledWidgets()                 | 98  |
| 14.11.3.11 showTemplate                        | 98  |
| 14.11.4 Member Data Documentation              | 98  |
| 14.11.4.1 addingNew_                           | 98  |
| 14.11.4.2 FT_                                  | 98  |
| 14.11.4.3 ui                                   | 98  |
| 14.12 EventTemplate Class Reference            | 99  |
| 14.12.1 Detailed Description                   | 99  |
| 14.12.2 Constructor & Destructor Documentation | 100 |
| 14.12.2.1 EventTemplate()                      | 100 |
| 14.12.3 Member Function Documentation          | 100 |
| 14.12.3.1 containsDate()                       | 100 |
| 14.12.3.2 containsPlace()                      | 100 |
| 14.12.3.3 getDescription()                     | 100 |
| 14.12.3.4 getId()                              | 100 |
| 14.12.3.5 getRoles()                           | 101 |
| 14.12.3.6 getTitle()                           | 101 |
| 14.12.3.7 hasMorePeopleInvolved()              | 101 |
| 14.12.3.8 hasRoles()                           | 101 |
| 14.12.3.9 readJson()                           | 101 |
| 14.12.3.10 setContainsDate()                   | 102 |
| 14.12.3.11 setContainsPlace()                  | 102 |
| 14.12.3.12 setDescription()                    | 102 |
| 14.12.3.13 setId()                             | 102 |
| 14.12.3.14 setMorePeopleInvolved()             | 103 |
| 14.12.3.15 setRoles()                          | 103 |
| 14.12.3.16 setTitle()                          | 103 |
| 14.12.3.17 str()                               | 103 |
| 14.12.4 Member Data Documentation              | 103 |
| 14.12.4.1 containsDate_                        | 103 |
| 14.12.4.2 containsPlace_                       | 104 |
| 14.12.4.3 description_                         | 104 |
| 14.12.4.4 id_                                  | 104 |
| 14.12.4.5 morePeopleInvolved_                  | 104 |
| 14.12.4.6 roles_                               | 104 |
| 14.12.4.7 title_                               | 104 |
| 14.13 ExportProjectDialog Class Reference      | 105 |
| 14.13.1 Detailed Description                   | 105 |
| 14.13.2 Constructor & Destructor Documentation | 105 |
| 14.13.2.1 ExportProjectDialog()                | 105 |

|  |     |
|--|-----|
| 14.13.2.2 ~ExportProjectDialog()               | 106 |
| 14.13.3 Member Function Documentation          | 106 |
| 14.13.3.1 checkPerson                          | 106 |
| 14.13.3.2 exportProject                        | 106 |
| 14.13.4 Member Data Documentation              | 106 |
| 14.13.4.1 FT_                                  | 106 |
| 14.13.4.2 ui                                   | 106 |
| 14.14 ExportTemplatesDialog Class Reference    | 107 |
| 14.14.1 Detailed Description                   | 107 |
| 14.14.2 Constructor & Destructor Documentation | 107 |
| 14.14.2.1 ExportTemplatesDialog()              | 107 |
| 14.14.2.2 ~ExportTemplatesDialog()             | 108 |
| 14.14.3 Member Function Documentation          | 108 |
| 14.14.3.1 exportTemplates                      | 108 |
| 14.14.4 Member Data Documentation              | 108 |
| 14.14.4.1 FT_                                  | 108 |
| 14.14.4.2 ui                                   | 108 |
| 14.15 FamilyTree Class Reference               | 108 |
| 14.15.1 Detailed Description                   | 111 |
| 14.15.2 Constructor & Destructor Documentation | 111 |
| 14.15.2.1 FamilyTree()                         | 111 |
| 14.15.3 Member Function Documentation          | 112 |
| 14.15.3.1 addEvent() [1/2]                     | 112 |
| 14.15.3.2 addEvent() [2/2]                     | 112 |
| 14.15.3.3 addFile() [1/2]                      | 112 |
| 14.15.3.4 addFile() [2/2]                      | 112 |
| 14.15.3.5 addPerson() [1/2]                    | 113 |
| 14.15.3.6 addPerson() [2/2]                    | 113 |
| 14.15.3.7 addRelation() [1/2]                  | 113 |
| 14.15.3.8 addRelation() [2/2]                  | 114 |
| 14.15.3.9 append()                             | 114 |
| 14.15.3.10 checkFileConsistence()              | 114 |
| 14.15.3.11 checkFileTypeConsistence()          | 114 |
| 14.15.3.12 clear()                             | 115 |
| 14.15.3.13 clearProjectPaths()                 | 115 |
| 14.15.3.14 copyFile()                          | 115 |
| 14.15.3.15 createDatabase()                    | 115 |
| 14.15.3.16 createEmptyFile()                   | 115 |
| 14.15.3.17 createResourcesDir()                | 116 |
| 14.15.3.18 enableHelpOnStartup()               | 116 |
| 14.15.3.19 exportProject()                     | 116 |
| 14.15.3.20 exportTemplates()                   | 117 |

|            |                          |     |
|------------|--------------------------|-----|
| 14.15.3.21 | getEvent()               | 117 |
| 14.15.3.22 | getEvents()              | 117 |
| 14.15.3.23 | getFile()                | 117 |
| 14.15.3.24 | getFileAbsolutePath()    | 118 |
| 14.15.3.25 | getFiles()               | 118 |
| 14.15.3.26 | getGeneralOrphaFiles()   | 118 |
| 14.15.3.27 | getMainPerson()          | 119 |
| 14.15.3.28 | getOrphanFiles()         | 119 |
| 14.15.3.29 | getParentSuggestions()   | 119 |
| 14.15.3.30 | getPartnersSuggestions() | 120 |
| 14.15.3.31 | getPerson()              | 120 |
| 14.15.3.32 | getPersons()             | 120 |
| 14.15.3.33 | getProjectPaths()        | 120 |
| 14.15.3.34 | getRelation()            | 121 |
| 14.15.3.35 | getRelations()           | 121 |
| 14.15.3.36 | getRelationSuggestions() | 121 |
| 14.15.3.37 | getSettings()            | 121 |
| 14.15.3.38 | getSiblingsSuggestions() | 122 |
| 14.15.3.39 | getUrlFile()             | 122 |
| 14.15.3.40 | importProject()          | 122 |
| 14.15.3.41 | importRootDir()          | 123 |
| 14.15.3.42 | importTemplates()        | 123 |
| 14.15.3.43 | isDirectorySet()         | 123 |
| 14.15.3.44 | isSaved()                | 123 |
| 14.15.3.45 | loadDatabaseFile()       | 124 |
| 14.15.3.46 | log()                    | 124 |
| 14.15.3.47 | openDatabase() [1/2]     | 124 |
| 14.15.3.48 | openDatabase() [2/2]     | 125 |
| 14.15.3.49 | openSingleFileDatabase() | 125 |
| 14.15.3.50 | printCss()               | 125 |
| 14.15.3.51 | printEvents()            | 125 |
| 14.15.3.52 | printFiles()             | 126 |
| 14.15.3.53 | printHtml()              | 126 |
| 14.15.3.54 | printHtmlPerson()        | 126 |
| 14.15.3.55 | printMainPersonHtml()    | 127 |
| 14.15.3.56 | printPeople()            | 127 |
| 14.15.3.57 | printRelations()         | 127 |
| 14.15.3.58 | promoteRelation()        | 127 |
| 14.15.3.59 | readJsonEvents()         | 128 |
| 14.15.3.60 | readJsonFiles()          | 128 |
| 14.15.3.61 | readJsonPeople()         | 128 |
| 14.15.3.62 | readJsonRelations()      | 128 |



|   |     |
|---|-----|
| 14.15.3.63 removeBackup()                     | 129 |
| 14.15.3.64 removeEvent()                      | 129 |
| 14.15.3.65 removeEventTemplate()              | 129 |
| 14.15.3.66 removeGeneralOrphanFile()          | 129 |
| 14.15.3.67 removeOrphanFiles()                | 130 |
| 14.15.3.68 removePerson()                     | 130 |
| 14.15.3.69 removeRelation()                   | 130 |
| 14.15.3.70 removeRelationTemplate()           | 130 |
| 14.15.3.71 renameFile()                       | 130 |
| 14.15.3.72 restoreBackup()                    | 131 |
| 14.15.3.73 setMainPerson() [1/2]              | 131 |
| 14.15.3.74 setMainPerson() [2/2]              | 131 |
| 14.15.3.75 setRelation()                      | 132 |
| 14.15.3.76 setUnsaved()                       | 132 |
| 14.15.3.77 showHelpOnStartup()                | 132 |
| 14.15.3.78 storeDatabase()                    | 132 |
| 14.15.3.79 storeFileDatabase()                | 133 |
| 14.15.3.80 templatesBasedOnEventTemplate()    | 134 |
| 14.15.3.81 templatesBasedOnRelationTemplate() | 134 |
| 14.15.3.82 updateEventsWithTemplate()         | 134 |
| 14.15.3.83 writeConfig()                      | 135 |
| 14.15.4 Member Data Documentation             | 135 |
| 14.15.4.1 allEvents_                          | 135 |
| 14.15.4.2 allFiles_                           | 135 |
| 14.15.4.3 allMedia_                           | 135 |
| 14.15.4.4 allNotes_                           | 135 |
| 14.15.4.5 allPersons_                         | 135 |
| 14.15.4.6 allRelations_                       | 136 |
| 14.15.4.7 event_index_                        | 136 |
| 14.15.4.8 file_index_                         | 136 |
| 14.15.4.9 mainPerson_                         | 136 |
| 14.15.4.10 media_index_                       | 136 |
| 14.15.4.11 note_index_                        | 136 |
| 14.15.4.12 openHelp                           | 136 |
| 14.15.4.13 parser_                            | 136 |
| 14.15.4.14 person_index_                      | 137 |
| 14.15.4.15 projectPaths                       | 137 |
| 14.15.4.16 relation_index_                    | 137 |
| 14.15.4.17 savedChanges_                      | 137 |
| 14.15.4.18 settings_                          | 137 |
| 14.16 File Class Reference                    | 137 |
| 14.16.1 Detailed Description                  | 138 |

|  |     |
|--|-----|
| 14.16.2 Constructor & Destructor Documentation | 138 |
| 14.16.2.1 File()                               | 138 |
| 14.16.3 Member Function Documentation          | 138 |
| 14.16.3.1 getFilename()                        | 138 |
| 14.16.3.2 getId()                              | 139 |
| 14.16.3.3 getRealName()                        | 139 |
| 14.16.3.4 readJson()                           | 139 |
| 14.16.3.5 setFilename()                        | 139 |
| 14.16.3.6 setId()                              | 139 |
| 14.16.4 Member Data Documentation              | 141 |
| 14.16.4.1 filename_                            | 141 |
| 14.16.4.2 id_                                  | 141 |
| 14.17 FileTreeItem Class Reference             | 141 |
| 14.17.1 Detailed Description                   | 142 |
| 14.17.2 Constructor & Destructor Documentation | 142 |
| 14.17.2.1 FileTreeItem() [1/2]                 | 142 |
| 14.17.2.2 FileTreeItem() [2/2]                 | 142 |
| 14.17.3 Member Function Documentation          | 144 |
| 14.17.3.1 addFile()                            | 144 |
| 14.17.3.2 addSubDrive()                        | 144 |
| 14.17.3.3 getDrive()                           | 144 |
| 14.17.3.4 getId()                              | 145 |
| 14.17.3.5 isDirectory()                        | 145 |
| 14.17.3.6 removeFile()                         | 145 |
| 14.17.3.7 removeFolder()                       | 145 |
| 14.17.3.8 rename()                             | 145 |
| 14.17.4 Member Data Documentation              | 146 |
| 14.17.4.1 drive_                               | 146 |
| 14.17.4.2 fileId_                              | 146 |
| 14.17.4.3 FT_                                  | 146 |
| 14.17.4.4 isDir_                               | 146 |
| 14.17.4.5 type_                                | 146 |
| 14.18 GraphicsView Class Reference             | 147 |
| 14.18.1 Detailed Description                   | 147 |
| 14.18.2 Constructor & Destructor Documentation | 147 |
| 14.18.2.1 GraphicsView()                       | 147 |
| 14.18.3 Member Function Documentation          | 147 |
| 14.18.3.1 wheelEvent()                         | 147 |
| 14.19 HelpDialog Class Reference               | 148 |
| 14.19.1 Detailed Description                   | 148 |
| 14.19.2 Constructor & Destructor Documentation | 148 |
| 14.19.2.1 HelpDialog()                         | 148 |

|  |     |
|--|-----|
| 14.19.2.2 ~HelpDialog()                        | 149 |
| 14.19.3 Member Function Documentation          | 149 |
| 14.19.3.1 openHelp                             | 149 |
| 14.19.4 Member Data Documentation              | 149 |
| 14.19.4.1 FT_                                  | 149 |
| 14.19.4.2 ui                                   | 149 |
| 14.20 HtmlExport Class Reference               | 150 |
| 14.20.1 Detailed Description                   | 150 |
| 14.20.2 Constructor & Destructor Documentation | 150 |
| 14.20.2.1 HtmlExport()                         | 150 |
| 14.20.2.2 ~HtmlExport()                        | 151 |
| 14.20.3 Member Function Documentation          | 151 |
| 14.20.3.1 allAboveGenerations                  | 151 |
| 14.20.3.2 allBelowGenerations                  | 151 |
| 14.20.3.3 exportAllHtml                        | 151 |
| 14.20.3.4 exportMainHtml                       | 152 |
| 14.20.4 Member Data Documentation              | 152 |
| 14.20.4.1 FT_                                  | 152 |
| 14.20.4.2 ui                                   | 152 |
| 14.21 MainWindow Class Reference               | 152 |
| 14.21.1 Detailed Description                   | 158 |
| 14.21.2 Constructor & Destructor Documentation | 158 |
| 14.21.2.1 MainWindow()                         | 158 |
| 14.21.2.2 ~MainWindow()                        | 159 |
| 14.21.3 Member Function Documentation          | 159 |
| 14.21.3.1 addGeneralFile()                     | 159 |
| 14.21.3.2 addGeneralFolder()                   | 159 |
| 14.21.3.3 addNewEvent                          | 159 |
| 14.21.3.4 addNewFile                           | 159 |
| 14.21.3.5 addNewFileFolder                     | 160 |
| 14.21.3.6 addNewMedia                          | 160 |
| 14.21.3.7 addNewMediaFolder                    | 160 |
| 14.21.3.8 addNewNote                           | 160 |
| 14.21.3.9 addNewNoteFolder                     | 160 |
| 14.21.3.10 addNewPerson                        | 160 |
| 14.21.3.11 addNewRelation                      | 160 |
| 14.21.3.12 allAboveGenerations                 | 160 |
| 14.21.3.13 allBelowGenerations                 | 161 |
| 14.21.3.14 changeGenerationsDown               | 161 |
| 14.21.3.15 changeGenerationsUp                 | 161 |
| 14.21.3.16 checkStoreRefreshPersonsInfo        | 161 |
| 14.21.3.17 clearProjectFilter                  | 162 |

|  |     |
|--|-----|
| 14.21.3.18 clearProjectPaths . . . . .                 | 162 |
| 14.21.3.19 clearUi() . . . . .                         | 162 |
| 14.21.3.20 closeEvent() . . . . .                      | 162 |
| 14.21.3.21 connectAllSlots() . . . . .                 | 162 |
| 14.21.3.22 disableFileFolderPrompt . . . . .           | 162 |
| 14.21.3.23 disableMediaFolderPrompt . . . . .          | 162 |
| 14.21.3.24 disableNoteFolderPrompt . . . . .           | 163 |
| 14.21.3.25 drawCloseContainer() . . . . .              | 163 |
| 14.21.3.26 drawCloseFamily() . . . . .                 | 164 |
| 14.21.3.27 drawFamilyTree() . . . . .                  | 164 |
| 14.21.3.28 drawGeneralGenerations() . . . . .          | 164 |
| 14.21.3.29 drawOlderGenerations() . . . . .            | 165 |
| 14.21.3.30 drawPartner() . . . . .                     | 165 |
| 14.21.3.31 drawYoungerGenerations() . . . . .          | 166 |
| 14.21.3.32 editEvent [1/2] . . . . .                   | 166 |
| 14.21.3.33 editEvent() [2/2] . . . . .                 | 166 |
| 14.21.3.34 editRelation [1/2] . . . . .                | 167 |
| 14.21.3.35 editRelation() [2/2] . . . . .              | 167 |
| 14.21.3.36 editTag . . . . .                           | 167 |
| 14.21.3.37 enableFileFolderPrompt . . . . .            | 167 |
| 14.21.3.38 enableGeneralFolderPrompt() . . . . .       | 168 |
| 14.21.3.39 enableGeneralFolderRenamePrompt() . . . . . | 168 |
| 14.21.3.40 enableMediaFolderPrompt . . . . .           | 168 |
| 14.21.3.41 enableNoteFolderPrompt . . . . .            | 168 |
| 14.21.3.42 exportCloseFamilyDiagram() . . . . .        | 168 |
| 14.21.3.43 exportCloseFamilyDiagramDialog . . . . .    | 169 |
| 14.21.3.44 exportGeneralDiagram() . . . . .            | 169 |
| 14.21.3.45 exportHtml . . . . .                        | 169 |
| 14.21.3.46 exportMainHtml . . . . .                    | 169 |
| 14.21.3.47 exportProject . . . . .                     | 169 |
| 14.21.3.48 exportTemplates . . . . .                   | 170 |
| 14.21.3.49 exportTreeDiagram() . . . . .               | 170 |
| 14.21.3.50 exportTreeDiagramDialog . . . . .           | 170 |
| 14.21.3.51 filterProjectItems . . . . .                | 170 |
| 14.21.3.52 findPerson . . . . .                        | 170 |
| 14.21.3.53 fitTreeInView . . . . .                     | 170 |
| 14.21.3.54 generalGenerationNumber() . . . . .         | 171 |
| 14.21.3.55 importProject . . . . .                     | 171 |
| 14.21.3.56 importRootDir . . . . .                     | 171 |
| 14.21.3.57 importTemplates . . . . .                   | 171 |
| 14.21.3.58 initializeProjectSubView() . . . . .        | 171 |
| 14.21.3.59 initializeProjectView() . . . . .           | 172 |

|   |     |
|---|-----|
| 14.21.3.60 interactWithProjectItem . . . . .          | 172 |
| 14.21.3.61 loadSettings() . . . . .                   | 172 |
| 14.21.3.62 olderGenerationNumber() . . . . .          | 172 |
| 14.21.3.63 openBirthDateDialog . . . . .              | 173 |
| 14.21.3.64 openCustomDateDialog() . . . . .           | 173 |
| 14.21.3.65 openDeathDateDialog . . . . .              | 173 |
| 14.21.3.66 openEventTemplatesDialog . . . . .         | 173 |
| 14.21.3.67 openExistingProject . . . . .              | 173 |
| 14.21.3.68 openExternalFile() . . . . .               | 173 |
| 14.21.3.69 openGeneralFile() . . . . .                | 174 |
| 14.21.3.70 openGeneralProject() . . . . .             | 174 |
| 14.21.3.71 openGraphicSettings . . . . .              | 174 |
| 14.21.3.72 openNewFamilyTreeProject . . . . .         | 174 |
| 14.21.3.73 openRelationTemplatesDialog . . . . .      | 174 |
| 14.21.3.74 openSelectedFile . . . . .                 | 175 |
| 14.21.3.75 openSelectedMedia . . . . .                | 175 |
| 14.21.3.76 openSelectedNote . . . . .                 | 175 |
| 14.21.3.77 promoteRelation . . . . .                  | 175 |
| 14.21.3.78 refreshDates . . . . .                     | 175 |
| 14.21.3.79 refreshEventTab . . . . .                  | 176 |
| 14.21.3.80 refreshGraphics . . . . .                  | 176 |
| 14.21.3.81 refreshPersonsInfoTab() . . . . .          | 176 |
| 14.21.3.82 refreshProjectEventsView() . . . . .       | 176 |
| 14.21.3.83 refreshProjectFilesView() . . . . .        | 176 |
| 14.21.3.84 refreshProjectGeneralFilesView() . . . . . | 177 |
| 14.21.3.85 refreshProjectMediaView() . . . . .        | 177 |
| 14.21.3.86 refreshProjectNoteView() . . . . .         | 177 |
| 14.21.3.87 refreshProjectRelationsView() . . . . .    | 178 |
| 14.21.3.88 refreshProjectView . . . . .               | 178 |
| 14.21.3.89 refreshRelationTab . . . . .               | 178 |
| 14.21.3.90 refreshTabsWithFiles . . . . .             | 178 |
| 14.21.3.91 refreshTagsList . . . . .                  | 178 |
| 14.21.3.92 refreshUi() . . . . .                      | 178 |
| 14.21.3.93 removeEvent . . . . .                      | 178 |
| 14.21.3.94 removeFileFolder . . . . .                 | 179 |
| 14.21.3.95 removeGeneralFolder() . . . . .            | 179 |
| 14.21.3.96 removeMediaFolder . . . . .                | 179 |
| 14.21.3.97 removeNoteFolder . . . . .                 | 179 |
| 14.21.3.98 removePerson . . . . .                     | 179 |
| 14.21.3.99 removeRelation . . . . .                   | 179 |
| 14.21.3.100 removeTag . . . . .                       | 179 |
| 14.21.3.101 renameFile . . . . .                      | 180 |

|   |     |
|---|-----|
| 14.21.3.102 renameGeneralFileItem()     | 180 |
| 14.21.3.103 renameMedia                 | 180 |
| 14.21.3.104 renameNote                  | 180 |
| 14.21.3.105 saveDatabase                | 180 |
| 14.21.3.106 saveGraphicsSetting         | 180 |
| 14.21.3.107 savePersonsInfo()           | 181 |
| 14.21.3.108 selectOlderGenerationOnly   | 181 |
| 14.21.3.109 selectSameGenerationOnly    | 181 |
| 14.21.3.110 selectYoungerGenerationOnly | 181 |
| 14.21.3.111 setEnabledDeathAtributes    | 181 |
| 14.21.3.112 setEventTabVisible          | 182 |
| 14.21.3.113 setFileTabVisible           | 182 |
| 14.21.3.114 setGeneralTabVisible        | 182 |
| 14.21.3.115 setMainPerson               | 182 |
| 14.21.3.116 setMediaTabVisible          | 182 |
| 14.21.3.117 setNoteTabVisible           | 182 |
| 14.21.3.118 setNumberOfGenerations()    | 183 |
| 14.21.3.119 setRelationTabVisible       | 183 |
| 14.21.3.120 setTabEast                  | 183 |
| 14.21.3.121 setTabNorth                 | 183 |
| 14.21.3.122 setTabSouth                 | 183 |
| 14.21.3.123 setTabWest                  | 183 |
| 14.21.3.124 setUiElements()             | 184 |
| 14.21.3.125 showGeneralFiles()          | 184 |
| 14.21.3.126 showHelp                    | 184 |
| 14.21.3.127 showRenameFile              | 184 |
| 14.21.3.128 showRenameMedia             | 184 |
| 14.21.3.129 showRenameNote              | 184 |
| 14.21.3.130 splitterChanged             | 185 |
| 14.21.3.131 switchNewTreeProject        | 185 |
| 14.21.3.132 youngerGenerationNumber()   | 185 |
| 14.21.3.133 zoomIn                      | 185 |
| 14.21.3.134 zoomOut                     | 185 |
| 14.21.4 Member Data Documentation       | 185 |
| 14.21.4.1 borderRadius_                 | 185 |
| 14.21.4.2 font_                         | 186 |
| 14.21.4.3 FT                            | 186 |
| 14.21.4.4 genSizeDown_                  | 186 |
| 14.21.4.5 genSizeUp_                    | 186 |
| 14.21.4.6 highlightedColor_             | 186 |
| 14.21.4.7 linePen_                      | 186 |
| 14.21.4.8 probandColor_                 | 186 |

|  |     |
|--|-----|
| 14.21.4.9 promotedColor_ . . . . .                       | 186 |
| 14.21.4.10 scene . . . . .                               | 187 |
| 14.21.4.11 sizeX_ . . . . .                              | 187 |
| 14.21.4.12 sizeY_ . . . . .                              | 187 |
| 14.21.4.13 standardColor_ . . . . .                      | 187 |
| 14.21.4.14 textPen_ . . . . .                            | 187 |
| 14.21.4.15 treeScene . . . . .                           | 187 |
| 14.21.4.16 treeView . . . . .                            | 187 |
| 14.21.4.17 ui . . . . .                                  | 188 |
| 14.22 Parser Class Reference . . . . .                   | 188 |
| 14.22.1 Detailed Description . . . . .                   | 189 |
| 14.22.2 Constructor & Destructor Documentation . . . . . | 189 |
| 14.22.2.1 Parser() . . . . .                             | 189 |
| 14.22.3 Member Function Documentation . . . . .          | 189 |
| 14.22.3.1 checkFileConsistency() . . . . .               | 189 |
| 14.22.3.2 clear() . . . . .                              | 189 |
| 14.22.3.3 containsBackup() . . . . .                     | 190 |
| 14.22.3.4 containsBackupFile() . . . . .                 | 190 |
| 14.22.3.5 copyFile() . . . . .                           | 190 |
| 14.22.3.6 createAllDirs() . . . . .                      | 191 |
| 14.22.3.7 createEmptyFile() . . . . .                    | 191 |
| 14.22.3.8 createJsonFile() . . . . .                     | 191 |
| 14.22.3.9 createResourcesDir() . . . . .                 | 191 |
| 14.22.3.10 getAbsolutePath() . . . . .                   | 192 |
| 14.22.3.11 getFilePath() . . . . .                       | 192 |
| 14.22.3.12 isRootDirectorySet() . . . . .                | 192 |
| 14.22.3.13 log() . . . . .                               | 193 |
| 14.22.3.14 makeNewDatabase() . . . . .                   | 193 |
| 14.22.3.15 readConfig() . . . . .                        | 193 |
| 14.22.3.16 readJSONFile() . . . . .                      | 193 |
| 14.22.3.17 removeBackup() . . . . .                      | 194 |
| 14.22.3.18 removeFile() . . . . .                        | 194 |
| 14.22.3.19 renameFile() . . . . .                        | 194 |
| 14.22.3.20 restoreBackup() . . . . .                     | 194 |
| 14.22.3.21 setDatabase() . . . . .                       | 195 |
| 14.22.3.22 writeConfig() [1/2] . . . . .                 | 195 |
| 14.22.3.23 writeConfig() [2/2] . . . . .                 | 195 |
| 14.22.3.24 writeHtml() [1/2] . . . . .                   | 196 |
| 14.22.3.25 writeHtml() [2/2] . . . . .                   | 196 |
| 14.22.3.26 writeJSON() . . . . .                         | 196 |
| 14.22.4 Member Data Documentation . . . . .              | 196 |
| 14.22.4.1 configPath_ . . . . .                          | 196 |

|  |     |
|--|-----|
| 14.22.4.2 root_ . . . . .                                | 197 |
| 14.23 Person Class Reference . . . . .                   | 197 |
| 14.23.1 Detailed Description . . . . .                   | 199 |
| 14.23.2 Constructor & Destructor Documentation . . . . . | 199 |
| 14.23.2.1 Person() . . . . .                             | 199 |
| 14.23.3 Member Function Documentation . . . . .          | 199 |
| 14.23.3.1 addEvent() . . . . .                           | 199 |
| 14.23.3.2 addRelation() . . . . .                        | 200 |
| 14.23.3.3 addTag() . . . . .                             | 200 |
| 14.23.3.4 birthDate() . . . . .                          | 200 |
| 14.23.3.5 deathDate() . . . . .                          | 200 |
| 14.23.3.6 existsTag() . . . . .                          | 200 |
| 14.23.3.7 exportJson() . . . . .                         | 201 |
| 14.23.3.8 getAfterTitle() . . . . .                      | 201 |
| 14.23.3.9 getBirthDate() . . . . .                       | 201 |
| 14.23.3.10 getBirthPlace() . . . . .                     | 202 |
| 14.23.3.11 getDeathDate() . . . . .                      | 202 |
| 14.23.3.12 getDeathPlace() . . . . .                     | 202 |
| 14.23.3.13 getEvents() . . . . .                         | 202 |
| 14.23.3.14 getFather() . . . . .                         | 202 |
| 14.23.3.15 getFilesRoot() . . . . .                      | 202 |
| 14.23.3.16 getFilesRootPointer() . . . . .               | 203 |
| 14.23.3.17 getFrontTitle() . . . . .                     | 203 |
| 14.23.3.18 getFullName() . . . . .                       | 203 |
| 14.23.3.19 getGender() . . . . .                         | 204 |
| 14.23.3.20 getId() . . . . .                             | 204 |
| 14.23.3.21 getMaidenName() . . . . .                     | 204 |
| 14.23.3.22 getMother() . . . . .                         | 204 |
| 14.23.3.23 getName() . . . . .                           | 204 |
| 14.23.3.24 getPartner() . . . . .                        | 205 |
| 14.23.3.25 getRelations() . . . . .                      | 205 |
| 14.23.3.26 getSurname() . . . . .                        | 205 |
| 14.23.3.27 getTags() . . . . .                           | 205 |
| 14.23.3.28 importJson() . . . . .                        | 205 |
| 14.23.3.29 isAlive() . . . . .                           | 206 |
| 14.23.3.30 printHtml() . . . . .                         | 206 |
| 14.23.3.31 promoteRelation() . . . . .                   | 206 |
| 14.23.3.32 readJson() . . . . .                          | 206 |
| 14.23.3.33 removeEvent() . . . . .                       | 207 |
| 14.23.3.34 removeRelation() . . . . .                    | 207 |
| 14.23.3.35 removeTag() . . . . .                         | 207 |
| 14.23.3.36 setAfterTitle() . . . . .                     | 207 |



|  |     |
|--|-----|
| 14.23.3.37 setBirthDate()                      | 208 |
| 14.23.3.38 setBirthPlace()                     | 208 |
| 14.23.3.39 setDeathDate()                      | 208 |
| 14.23.3.40 setDeathPlace()                     | 209 |
| 14.23.3.41 setFather()                         | 209 |
| 14.23.3.42 setFilesRoot()                      | 209 |
| 14.23.3.43 setFrontTitle()                     | 209 |
| 14.23.3.44 setGender()                         | 210 |
| 14.23.3.45 setId()                             | 210 |
| 14.23.3.46 setLives()                          | 210 |
| 14.23.3.47 setMaidenName()                     | 210 |
| 14.23.3.48 setMother()                         | 210 |
| 14.23.3.49 setName()                           | 211 |
| 14.23.3.50 setPartner()                        | 211 |
| 14.23.3.51 setSurname()                        | 211 |
| 14.23.3.52 str()                               | 211 |
| 14.23.3.53 updateSpecialRelation()             | 212 |
| 14.23.4 Member Data Documentation              | 213 |
| 14.23.4.1 dateOfBirth_                         | 213 |
| 14.23.4.2 dateOfDeath_                         | 213 |
| 14.23.4.3 events_                              | 213 |
| 14.23.4.4 father_                              | 213 |
| 14.23.4.5 gender_                              | 213 |
| 14.23.4.6 id_                                  | 213 |
| 14.23.4.7 lives_                               | 214 |
| 14.23.4.8 maidenName_                          | 214 |
| 14.23.4.9 mother_                              | 214 |
| 14.23.4.10 name_                               | 214 |
| 14.23.4.11 partner_                            | 214 |
| 14.23.4.12 placeOfBirth_                       | 214 |
| 14.23.4.13 placeOfDeath_                       | 214 |
| 14.23.4.14 relations_                          | 214 |
| 14.23.4.15 rootFileDrive_                      | 215 |
| 14.23.4.16 rootMediaDrive_                     | 215 |
| 14.23.4.17 rootNoteDrive_                      | 215 |
| 14.23.4.18 surname_                            | 215 |
| 14.23.4.19 tags_                               | 215 |
| 14.23.4.20 titleAfter_                         | 215 |
| 14.23.4.21 titleInFront_                       | 215 |
| 14.24 PersonsGraphicsItem Class Reference      | 216 |
| 14.24.1 Detailed Description                   | 217 |
| 14.24.2 Constructor & Destructor Documentation | 217 |

|  |     |
|--|-----|
| 14.24.2.1 PersonsGraphicsItem()                | 217 |
| 14.24.2.2 ~PersonsGraphicsItem()               | 218 |
| 14.24.3 Member Function Documentation          | 218 |
| 14.24.3.1 boundingRect()                       | 218 |
| 14.24.3.2 hoverEnterEvent()                    | 218 |
| 14.24.3.3 hoverLeaveEvent()                    | 218 |
| 14.24.3.4 itemClicked                          | 219 |
| 14.24.3.5 mousePressEvent()                    | 219 |
| 14.24.3.6 paint()                              | 219 |
| 14.24.3.7 shape()                              | 219 |
| 14.24.4 Member Data Documentation              | 220 |
| 14.24.4.1 bold_text_                           | 220 |
| 14.24.4.2 borderRadius_                        | 220 |
| 14.24.4.3 color_                               | 220 |
| 14.24.4.4 font_                                | 220 |
| 14.24.4.5 highlighted_                         | 220 |
| 14.24.4.6 hovered_                             | 220 |
| 14.24.4.7 pen_                                 | 220 |
| 14.24.4.8 person_                              | 221 |
| 14.24.4.9 sizex_                               | 221 |
| 14.24.4.10 sizey_                              | 221 |
| 14.24.4.11 textLines_                          | 221 |
| 14.24.4.12 textPen_                            | 221 |
| 14.24.4.13 x_                                  | 221 |
| 14.24.4.14 y_                                  | 221 |
| 14.25 PictureExport Class Reference            | 222 |
| 14.25.1 Detailed Description                   | 222 |
| 14.25.2 Constructor & Destructor Documentation | 222 |
| 14.25.2.1 PictureExport()                      | 222 |
| 14.25.2.2 ~PictureExport()                     | 223 |
| 14.25.3 Member Function Documentation          | 223 |
| 14.25.3.1 exportDiagram                        | 223 |
| 14.25.4 Member Data Documentation              | 223 |
| 14.25.4.1 closeDiagram_                        | 223 |
| 14.25.4.2 ui                                   | 223 |
| 14.26 ProjectItem Class Reference              | 223 |
| 14.26.1 Detailed Description                   | 224 |
| 14.26.2 Constructor & Destructor Documentation | 224 |
| 14.26.2.1 ProjectItem()                        | 224 |
| 14.26.3 Member Function Documentation          | 224 |
| 14.26.3.1 getId()                              | 224 |
| 14.26.3.2 type()                               | 225 |

|  |     |
|--|-----|
| 14.26.4 Member Data Documentation              | 225 |
| 14.26.4.1 id_                                  | 225 |
| 14.26.4.2 type_                                | 225 |
| 14.27 Relation Class Reference                 | 225 |
| 14.27.1 Detailed Description                   | 226 |
| 14.27.2 Constructor & Destructor Documentation | 226 |
| 14.27.2.1 Relation()                           | 226 |
| 14.27.3 Member Function Documentation          | 226 |
| 14.27.3.1 getFirstPerson()                     | 226 |
| 14.27.3.2 getId()                              | 226 |
| 14.27.3.3 getSecondPerson()                    | 227 |
| 14.27.3.4 getTemplate()                        | 227 |
| 14.27.3.5 getTheOtherPerson()                  | 227 |
| 14.27.3.6 getTheOtherPersonName()              | 227 |
| 14.27.3.7 getTitle()                           | 228 |
| 14.27.3.8 getTrait()                           | 228 |
| 14.27.3.9 isOlderGeneration()                  | 228 |
| 14.27.3.10 isParent()                          | 228 |
| 14.27.3.11 isSameGenration()                   | 229 |
| 14.27.3.12 isYoungerGeneration()               | 229 |
| 14.27.3.13 printHtml()                         | 229 |
| 14.27.3.14 readJson()                          | 230 |
| 14.27.3.15 setId()                             | 230 |
| 14.27.3.16 setPersons()                        | 230 |
| 14.27.3.17 setTemplate()                       | 230 |
| 14.27.3.18 str()                               | 231 |
| 14.27.4 Member Data Documentation              | 231 |
| 14.27.4.1 id_                                  | 231 |
| 14.27.4.2 person1_                             | 231 |
| 14.27.4.3 person2_                             | 231 |
| 14.27.4.4 settings_                            | 231 |
| 14.27.4.5 template_                            | 232 |
| 14.28 RelationDialog Class Reference           | 232 |
| 14.28.1 Detailed Description                   | 233 |
| 14.28.2 Constructor & Destructor Documentation | 233 |
| 14.28.2.1 RelationDialog()                     | 233 |
| 14.28.2.2 ~RelationDialog()                    | 233 |
| 14.28.3 Member Function Documentation          | 233 |
| 14.28.3.1 chooseFirstPerson                    | 233 |
| 14.28.3.2 chooseSecondPerson                   | 234 |
| 14.28.3.3 savePerson                           | 234 |
| 14.28.3.4 saveRelation                         | 234 |

|  |     |
|--|-----|
| 14.28.3.5 showTemplateDetails . . . . .                  | 234 |
| 14.28.3.6 switchPersons . . . . .                        | 234 |
| 14.28.4 Member Data Documentation . . . . .              | 234 |
| 14.28.4.1 FT_ . . . . .                                  | 234 |
| 14.28.4.2 person1_ . . . . .                             | 234 |
| 14.28.4.3 person2_ . . . . .                             | 235 |
| 14.28.4.4 rel_ . . . . .                                 | 235 |
| 14.28.4.5 saveFirstPerson_ . . . . .                     | 235 |
| 14.28.4.6 ui . . . . .                                   | 235 |
| 14.29 RelationItem Class Reference . . . . .             | 235 |
| 14.29.1 Detailed Description . . . . .                   | 236 |
| 14.29.2 Constructor & Destructor Documentation . . . . . | 236 |
| 14.29.2.1 RelationItem() . . . . .                       | 236 |
| 14.29.3 Member Function Documentation . . . . .          | 236 |
| 14.29.3.1 getRelation() . . . . .                        | 236 |
| 14.29.3.2 refreshTexts() . . . . .                       | 236 |
| 14.29.4 Member Data Documentation . . . . .              | 237 |
| 14.29.4.1 FT_ . . . . .                                  | 237 |
| 14.29.4.2 relation_ . . . . .                            | 237 |
| 14.30 RelationSuggestion Struct Reference . . . . .      | 237 |
| 14.30.1 Detailed Description . . . . .                   | 237 |
| 14.30.2 Constructor & Destructor Documentation . . . . . | 238 |
| 14.30.2.1 RelationSuggestion() . . . . .                 | 238 |
| 14.30.3 Member Data Documentation . . . . .              | 239 |
| 14.30.3.1 lockedPerson . . . . .                         | 239 |
| 14.30.3.2 parent . . . . .                               | 239 |
| 14.30.3.3 suggestedPerson . . . . .                      | 239 |
| 14.30.3.4 symetrical . . . . .                           | 239 |
| 14.30.3.5 trait . . . . .                                | 239 |
| 14.31 RelationTemplate Class Reference . . . . .         | 240 |
| 14.31.1 Detailed Description . . . . .                   | 240 |
| 14.31.2 Constructor & Destructor Documentation . . . . . | 241 |
| 14.31.2.1 RelationTemplate() . . . . .                   | 241 |
| 14.31.3 Member Function Documentation . . . . .          | 241 |
| 14.31.3.1 getDescription() . . . . .                     | 241 |
| 14.31.3.2 getFirstName() . . . . .                       | 241 |
| 14.31.3.3 getGenerationDifference() . . . . .            | 241 |
| 14.31.3.4 getId() . . . . .                              | 241 |
| 14.31.3.5 getSecondName() . . . . .                      | 242 |
| 14.31.3.6 getTitle() . . . . .                           | 242 |
| 14.31.3.7 getTrait() . . . . .                           | 242 |
| 14.31.3.8 readJson() . . . . .                           | 242 |

|  |     |
|--|-----|
| 14.31.3.9 setDescription()                     | 242 |
| 14.31.3.10 setFirstName()                      | 243 |
| 14.31.3.11 setGenerationDifference()           | 243 |
| 14.31.3.12 setId()                             | 243 |
| 14.31.3.13 setSecondName()                     | 243 |
| 14.31.3.14 setTitle()                          | 244 |
| 14.31.3.15 setTrait()                          | 244 |
| 14.31.3.16 str()                               | 244 |
| 14.31.4 Member Data Documentation              | 244 |
| 14.31.4.1 description_                         | 244 |
| 14.31.4.2 firstName_                           | 244 |
| 14.31.4.3 generationDiff_                      | 245 |
| 14.31.4.4 id_                                  | 245 |
| 14.31.4.5 secondName_                          | 245 |
| 14.31.4.6 title_                               | 245 |
| 14.31.4.7 trait_                               | 245 |
| 14.32 RelationTemplateDialog Class Reference   | 245 |
| 14.32.1 Detailed Description                   | 246 |
| 14.32.2 Constructor & Destructor Documentation | 246 |
| 14.32.2.1 RelationTemplateDialog()             | 246 |
| 14.32.2.2 ~RelationTemplateDialog()            | 247 |
| 14.32.3 Member Function Documentation          | 247 |
| 14.32.3.1 addDefaults                          | 247 |
| 14.32.3.2 addNewTemplate                       | 247 |
| 14.32.3.3 clear()                              | 247 |
| 14.32.3.4 refreshListWidget()                  | 247 |
| 14.32.3.5 removeTemplate                       | 247 |
| 14.32.3.6 saveTemplate [1/2]                   | 247 |
| 14.32.3.7 saveTemplate() [2/2]                 | 247 |
| 14.32.3.8 setEnabledWidgets()                  | 248 |
| 14.32.3.9 showTemplate                         | 248 |
| 14.32.4 Member Data Documentation              | 248 |
| 14.32.4.1 addingNew_                           | 248 |
| 14.32.4.2 FT_                                  | 248 |
| 14.32.4.3 ui                                   | 248 |
| 14.33 Settings Class Reference                 | 249 |
| 14.33.1 Detailed Description                   | 250 |
| 14.33.2 Constructor & Destructor Documentation | 250 |
| 14.33.2.1 Settings()                           | 250 |
| 14.33.3 Member Function Documentation          | 250 |
| 14.33.3.1 addDefaultEvent()                    | 250 |
| 14.33.3.2 addDefaultEvents()                   | 251 |

|  |     |
|--|-----|
| 14.33.3.3 addDefaultRelation()                 | 251 |
| 14.33.3.4 addDefaultRelations()                | 251 |
| 14.33.3.5 addEventTemplate()                   | 251 |
| 14.33.3.6 addRelationTemplate()                | 252 |
| 14.33.3.7 addTag()                             | 252 |
| 14.33.3.8 clear()                              | 252 |
| 14.33.3.9 exportTemplates()                    | 252 |
| 14.33.3.10 getAppSettings()                    | 253 |
| 14.33.3.11 getEventTemplate()                  | 253 |
| 14.33.3.12 getEventTemplates()                 | 253 |
| 14.33.3.13 getGlobalMainPerson()               | 253 |
| 14.33.3.14 getRelationTemplate()               | 253 |
| 14.33.3.15 getRelationTemplates()              | 254 |
| 14.33.3.16 getTags()                           | 254 |
| 14.33.3.17 importTemplates()                   | 254 |
| 14.33.3.18 indexes()                           | 254 |
| 14.33.3.19 readJson()                          | 255 |
| 14.33.3.20 reIndex()                           | 255 |
| 14.33.3.21 removeEventTemplate()               | 255 |
| 14.33.3.22 removeRelationTemplate()            | 255 |
| 14.33.3.23 removeTag()                         | 257 |
| 14.33.3.24 setAppSettings()                    | 257 |
| 14.33.3.25 setGlobalMainPerson()               | 257 |
| 14.33.4 Member Data Documentation              | 257 |
| 14.33.4.1 appSettings_                         | 257 |
| 14.33.4.2 eventTemplateIndex_                  | 258 |
| 14.33.4.3 eventTemplates_                      | 258 |
| 14.33.4.4 globalMainPerson_                    | 258 |
| 14.33.4.5 relationsTemplates_                  | 258 |
| 14.33.4.6 relationTemplateIndex_               | 258 |
| 14.33.4.7 tags_                                | 258 |
| 14.34 SuggestionsDialog Class Reference        | 259 |
| 14.34.1 Detailed Description                   | 260 |
| 14.34.2 Constructor & Destructor Documentation | 260 |
| 14.34.2.1 SuggestionsDialog()                  | 260 |
| 14.34.2.2 ~SuggestionsDialog()                 | 260 |
| 14.34.3 Member Function Documentation          | 260 |
| 14.34.3.1 removeSuggestion                     | 260 |
| 14.34.3.2 save                                 | 260 |
| 14.34.3.3 selectedSuggestion                   | 260 |
| 14.34.3.4 swapPersons                          | 261 |
| 14.34.3.5 switchedTemplate                     | 261 |

|  |     |
|--|-----|
| 14.34.4 Member Data Documentation              | 261 |
| 14.34.4.1 fatherhoods                          | 261 |
| 14.34.4.2 FT_                                  | 261 |
| 14.34.4.3 motherhoods                          | 261 |
| 14.34.4.4 partnerships                         | 261 |
| 14.34.4.5 siblings                             | 262 |
| 14.34.4.6 ui                                   | 262 |
| 14.35 TagChooserDialog Class Reference         | 262 |
| 14.35.1 Detailed Description                   | 263 |
| 14.35.2 Constructor & Destructor Documentation | 263 |
| 14.35.2.1 TagChooserDialog()                   | 263 |
| 14.35.2.2 ~TagChooserDialog()                  | 263 |
| 14.35.3 Member Function Documentation          | 263 |
| 14.35.3.1 changeVisibleWidgets                 | 263 |
| 14.35.3.2 saveTag                              | 263 |
| 14.35.4 Member Data Documentation              | 264 |
| 14.35.4.1 FT_                                  | 264 |
| 14.35.4.2 newOne_                              | 264 |
| 14.35.4.3 ui                                   | 264 |
| 14.36 VirtualDrive Class Reference             | 264 |
| 14.36.1 Detailed Description                   | 265 |
| 14.36.2 Constructor & Destructor Documentation | 265 |
| 14.36.2.1 VirtualDrive() [1/2]                 | 265 |
| 14.36.2.2 VirtualDrive() [2/2]                 | 265 |
| 14.36.3 Member Function Documentation          | 265 |
| 14.36.3.1 addFile()                            | 265 |
| 14.36.3.2 addSubdrive() [1/2]                  | 266 |
| 14.36.3.3 addSubdrive() [2/2]                  | 266 |
| 14.36.3.4 copyDrive()                          | 266 |
| 14.36.3.5 getFiles()                           | 266 |
| 14.36.3.6 getName()                            | 267 |
| 14.36.3.7 getSubdrives()                       | 267 |
| 14.36.3.8 readJson()                           | 267 |
| 14.36.3.9 remove()                             | 267 |
| 14.36.3.10 removeFile()                        | 267 |
| 14.36.3.11 rename()                            | 269 |
| 14.36.3.12 usedFiles()                         | 269 |
| 14.36.4 Member Data Documentation              | 269 |
| 14.36.4.1 files_                               | 269 |
| 14.36.4.2 name_                                | 269 |
| 14.36.4.3 subdrives_                           | 269 |
| 14.37 VisualSettings Class Reference           | 270 |

|  |     |
|--|-----|
| 14.37.1 Detailed Description                   | 271 |
| 14.37.2 Constructor & Destructor Documentation | 271 |
| 14.37.2.1 VisualSettings()                     | 271 |
| 14.37.2.2 ~VisualSettings()                    | 272 |
| 14.37.3 Member Function Documentation          | 272 |
| 14.37.3.1 highlightedColor                     | 272 |
| 14.37.3.2 lineColor                            | 272 |
| 14.37.3.3 probandColor                         | 272 |
| 14.37.3.4 promotedColor                        | 272 |
| 14.37.3.5 refresh                              | 273 |
| 14.37.3.6 saveAndExit                          | 273 |
| 14.37.3.7 saveSettings                         | 273 |
| 14.37.3.8 standardColor                        | 273 |
| 14.37.3.9 textColor                            | 274 |
| 14.37.4 Member Data Documentation              | 274 |
| 14.37.4.1 borderRadius_                        | 274 |
| 14.37.4.2 font_                                | 274 |
| 14.37.4.3 highlightedColor_                    | 274 |
| 14.37.4.4 pen_                                 | 274 |
| 14.37.4.5 probandColor_                        | 274 |
| 14.37.4.6 promotedColor_                       | 274 |
| 14.37.4.7 scene                                | 275 |
| 14.37.4.8 sizeX_                               | 275 |
| 14.37.4.9 sizeY_                               | 275 |
| 14.37.4.10 standardColor_                      | 275 |
| 14.37.4.11 textPen_                            | 275 |
| 14.37.4.12 ui                                  | 275 |
| 14.38 WrappedDate Class Reference              | 275 |
| 14.38.1 Detailed Description                   | 276 |
| 14.38.2 Constructor & Destructor Documentation | 276 |
| 14.38.2.1 WrappedDate()                        | 276 |
| 14.38.3 Member Function Documentation          | 276 |
| 14.38.3.1 clear()                              | 276 |
| 14.38.3.2 getFirstDate()                       | 277 |
| 14.38.3.3 getSecondDate()                      | 277 |
| 14.38.3.4 getText()                            | 277 |
| 14.38.3.5 isExact()                            | 277 |
| 14.38.3.6 isUnknown()                          | 277 |
| 14.38.3.7 readJson()                           | 277 |
| 14.38.3.8 setFirstDate()                       | 278 |
| 14.38.3.9 setSecondDate()                      | 278 |
| 14.38.3.10 setText()                           | 278 |



|  |            |
|--|------------|
| 14.38.3.11 str()                         | 279        |
| 14.38.3.12 updateDate()                  | 279        |
| 14.38.4 Member Data Documentation        | 279        |
| 14.38.4.1 date1_                         | 279        |
| 14.38.4.2 date2_                         | 279        |
| 14.38.4.3 text_                          | 279        |
| <b>15 File Documentation</b>             | <b>281</b> |
| 15.1 core/config.cpp File Reference      | 281        |
| 15.1.1 Detailed Description              | 281        |
| 15.1.2 Function Documentation            | 281        |
| 15.1.2.1 operator<<() [1/5]              | 281        |
| 15.1.2.2 operator<<() [2/5]              | 282        |
| 15.1.2.3 operator<<() [3/5]              | 282        |
| 15.1.2.4 operator<<() [4/5]              | 282        |
| 15.1.2.5 operator<<() [5/5]              | 283        |
| 15.2 core/config.h File Reference        | 283        |
| 15.2.1 Detailed Description              | 284        |
| 15.2.2 Enumeration Type Documentation    | 284        |
| 15.2.2.1 TabPosition                     | 284        |
| 15.2.2.2 Trait                           | 285        |
| 15.2.3 Function Documentation            | 285        |
| 15.2.3.1 operator<<() [1/5]              | 285        |
| 15.2.3.2 operator<<() [2/5]              | 285        |
| 15.2.3.3 operator<<() [3/5]              | 286        |
| 15.2.3.4 operator<<() [4/5]              | 286        |
| 15.2.3.5 operator<<() [5/5]              | 286        |
| 15.3 config.h                            | 287        |
| 15.4 core/date.cpp File Reference        | 289        |
| 15.4.1 Detailed Description              | 289        |
| 15.4.2 Function Documentation            | 289        |
| 15.4.2.1 operator<<() [1/2]              | 289        |
| 15.4.2.2 operator<<() [2/2]              | 290        |
| 15.5 core/date.h File Reference          | 290        |
| 15.5.1 Detailed Description              | 290        |
| 15.5.2 Function Documentation            | 291        |
| 15.5.2.1 operator<<() [1/2]              | 291        |
| 15.5.2.2 operator<<() [2/2]              | 291        |
| 15.6 date.h                              | 291        |
| 15.7 core/family_tree.cpp File Reference | 292        |
| 15.7.1 Detailed Description              | 292        |
| 15.8 core/family_tree.h File Reference   | 292        |

|   |     |
|---|-----|
| 15.8.1 Detailed Description                     | 293 |
| 15.9 family_tree.h                              | 293 |
| 15.10 core/family_tree_items.cpp File Reference | 295 |
| 15.10.1 Detailed Description                    | 295 |
| 15.10.2 Function Documentation                  | 295 |
| 15.10.2.1 operator<<() [1/3]                    | 295 |
| 15.10.2.2 operator<<() [2/3]                    | 295 |
| 15.10.2.3 operator<<() [3/3]                    | 296 |
| 15.11 core/family_tree_items.h File Reference   | 296 |
| 15.11.1 Detailed Description                    | 297 |
| 15.11.2 Function Documentation                  | 297 |
| 15.11.2.1 operator<<() [1/3]                    | 297 |
| 15.11.2.2 operator<<() [2/3]                    | 297 |
| 15.11.2.3 operator<<() [3/3]                    | 298 |
| 15.12 family_tree_items.h                       | 298 |
| 15.13 core/file_parser.cpp File Reference       | 299 |
| 15.13.1 Detailed Description                    | 299 |
| 15.14 core/file_parser.h File Reference         | 300 |
| 15.14.1 Detailed Description                    | 301 |
| 15.15 file_parser.h                             | 301 |
| 15.16 core/person.cpp File Reference            | 302 |
| 15.16.1 Detailed Description                    | 302 |
| 15.16.2 Function Documentation                  | 302 |
| 15.16.2.1 operator<<() [1/2]                    | 302 |
| 15.16.2.2 operator<<() [2/2]                    | 302 |
| 15.16.2.3 printSizeTContainer()                 | 303 |
| 15.17 core/person.h File Reference              | 303 |
| 15.17.1 Detailed Description                    | 304 |
| 15.17.2 Enumeration Type Documentation          | 304 |
| 15.17.2.1 FileType                              | 304 |
| 15.17.2.2 Gender                                | 304 |
| 15.17.3 Function Documentation                  | 305 |
| 15.17.3.1 operator<<() [1/2]                    | 305 |
| 15.17.3.2 operator<<() [2/2]                    | 305 |
| 15.17.3.3 printSizeTContainer()                 | 305 |
| 15.18 person.h                                  | 306 |
| 15.19 core/strings.h File Reference             | 307 |
| 15.19.1 Detailed Description                    | 315 |
| 15.19.2 Variable Documentation                  | 315 |
| 15.19.2.1 EMPTY_STRING                          | 315 |
| 15.20 strings.h                                 | 315 |
| 15.21 graphics/dialogs.cpp File Reference       | 318 |

|  |     |
|--|-----|
| 15.21.1 Detailed Description . . . . .                       | 318 |
| 15.22 graphics/dialogs.h File Reference . . . . .            | 318 |
| 15.22.1 Detailed Description . . . . .                       | 319 |
| 15.23 dialogs.h . . . . .                                    | 320 |
| 15.24 graphics/graphics_items.cpp File Reference . . . . .   | 323 |
| 15.24.1 Detailed Description . . . . .                       | 324 |
| 15.24.2 Function Documentation . . . . .                     | 324 |
| 15.24.2.1 containsForb() . . . . .                           | 324 |
| 15.25 graphics/graphics_items.h File Reference . . . . .     | 324 |
| 15.25.1 Detailed Description . . . . .                       | 325 |
| 15.25.2 Enumeration Type Documentation . . . . .             | 325 |
| 15.25.2.1 ProjectItemType . . . . .                          | 325 |
| 15.25.3 Function Documentation . . . . .                     | 325 |
| 15.25.3.1 containsForb() . . . . .                           | 325 |
| 15.26 graphics_items.h . . . . .                             | 326 |
| 15.27 graphics/mainwindow.cpp File Reference . . . . .       | 327 |
| 15.27.1 Detailed Description . . . . .                       | 327 |
| 15.28 graphics/mainwindow.h File Reference . . . . .         | 328 |
| 15.28.1 Detailed Description . . . . .                       | 328 |
| 15.29 mainwindow.h . . . . .                                 | 328 |
| 15.30 graphics/mainwindow_slots.cpp File Reference . . . . . | 331 |
| 15.30.1 Detailed Description . . . . .                       | 331 |
| 15.31 graphics/resources/icons.md File Reference . . . . .   | 331 |
| 15.32 help texts/events.md File Reference . . . . .          | 331 |
| 15.33 help texts/files.md File Reference . . . . .           | 331 |
| 15.34 help texts/generic.md File Reference . . . . .         | 331 |
| 15.35 help texts/layout.md File Reference . . . . .          | 331 |
| 15.36 help texts/persons.md File Reference . . . . .         | 331 |
| 15.37 help texts/relations.md File Reference . . . . .       | 331 |
| 15.38 index.md File Reference . . . . .                      | 331 |
| 15.39 main.cpp File Reference . . . . .                      | 331 |
| 15.39.1 Detailed Description . . . . .                       | 332 |
| 15.39.2 Function Documentation . . . . .                     | 332 |
| 15.39.2.1 main() . . . . .                                   | 332 |

|              |            |
|--------------|------------|
| <b>Index</b> | <b>333</b> |
|--------------|------------|



# Chapter 1

## Rodoc

This is the main page for documentation of the program Rodoc.

The code is separated mainly into two parts:

1. Core
2. Graphics

Core is for working with the family tree and also for parsing files.

Graphics is just the graphical interface of the application.

This application uses `Jsoncpp library`, `Qt graphics library` with version 6.

Database is a group of `json` files and directories. It follows this predefined structure (`#` is for a comment):

```
[rootDir]      # Chosen by the user.
- Persons.json # All persons.
- Config.json  # Configuration and templates.
- Events.json  # Events.
- Relations.json # Relations
- Files.json   # Files metadata.
- Media.json   # Media metadata.
- Notes.json   # Notes metadata.
- Media        # Directory for media.
- [files]
- Notes        # Directory for notes.
- [file]
- Files        # Directory for other files.
- [files]
```

Also `Files`, `Notes` and `Media` are internally implemented the same way. The main difference is for the end user.



## Chapter 2

# icons

Převzaté z [link](#).

Hlavní stránka je [tady](#).





## Chapter 3

# Events

**Event** entity allows to describe a particular event and associate it with the following pieces of information:

- **Date** (e.g., *20. 7. 2023*)
- **Place** (e.g., *Prague, Czech Republic*)
- **List of an arbitrary number of persons** acting as participants

In order to make the concept of events more generic and not limit ourselves to certain hard-wired event types only, each event is derived from a particular **event template**.

### 3.1 Event Templates

Each template has the following characteristics:

- **Title** (e.g., *Wedding*)
- **Description** (e.g., *A social event mainly between soon to be wife and husband.*)
- Flag activating a **date** (e.g., *enabled*)
- Flag activating a **place** (e.g., *disabled*)
- Arbitrary number of **roles** allowing to distinguish individual **participating persons** (e.g., *groom, bride, ...*)
- Flag allowing **additional participating persons** without any roles assigned

### 3.2 Predefined Templates

**Event** templates must be created first before they can be used. While it is possible to introduce specific user-defined templates, several widely used templates were already predefined and are available for import:

- **Wedding**
- **Divorce**
- **Funeral**
- **Graduation**



# Chapter 4

## Files, Notes & Media

Each person can have an unlimited number of **files**, **media**, and **notes**. To organize files easily, every **person** has a separate virtual directory for each file type. When a file is already added to the database, it can be **shared** among other persons without making **duplicates**.

### 4.1 Virtual Directories

Every virtual directory can be organized individually, just like in every file manager.

- **Renaming** a file or a directory. When renaming a file, it will be renamed for all persons, but the names of directories are not shared
- **Deleting** a file or a directory
- **Creating** a new directory
- **Adding** new files

Keep in mind that all the files need to be in a directory, thus there is always the **root** directory that cannot be deleted, but it can be renamed.

### 4.2 Adding Files

We have three options for adding files:

1. Choosing a file that is **already in the database**
2. Making a **copy** of an already existing file from a computer
3. Creating a completely **new and empty file**

To keep the project internally consistent and clean, files are **permanently removed** upon saving the project if it happens that no **person** has access to them.



## Chapter 5

# Welcome to rodoc

This is a brief introduction to this application for organizing family trees. Each **family tree** is saved and organized in a **project**.

### 5.1 Project

Each **project** is stored in a specific directory on a filesystem, assuming that the entire subtree of the project root directory should not be used for different purposes.

Various kinds of entities can be used throughout the project to express all the intended information related not just to persons and their relations.

### 5.2 Persons

Each family tree primarily contains a list of **persons**, each having basic information about themselves, such as a **name**, **surname**, etc. Each person can also have multiple user-defined **tags**.

### 5.3 Relations

To bind two particular **persons** together, we use the concept of a **relation** to record their real-world interconnection.

### 5.4 Events

**Persons** may also attend various types of events, such as weddings, allowing these to be associated with a **place**, **date**, as well as allowing the persons to participate in specific roles.

### 5.5 Files

Finally, each **person** can have an unlimited number of physical files of arbitrary types, logically divided into three groups of **files**, **media**, and **notes**.



# Chapter 6

## Layout

The **main window** has its **menu** and two separate **panels**.

### 6.1 Menu

The **menu** is for using the app functions.

- Changing the **settings**
- Saving and opening **projects**
- Editing **templates**
- Making **exports**
- **Importing** data

### 6.2 Panels

The left panel displays the information regarding the active person or so-called **proband**, while the right panel serves for the **family tree** visualization.

#### 6.2.1 Left Panel

The left panel contains the name of the **proband** at the top, and next to that options to create a new person or delete the current one. Under that, there are several tabs:

- **General** serves for changing the default **proband** information and **tags**
- Tabs **relations**, **events**, **files**, **media**, and **notes** permits organizing the same-named entities

#### 6.2.2 Right Panel

The right panel is also internally organized into three tabs where every view is interactive.

1. **Project overview** is a list of all **entities**, which can be filtered.
2. **Close family** is a diagram of all persons in relation to the **proband**. These persons are separated into three generations.
3. **Tree view** is a diagram of the **family tree** showing the so-called *hourglass* shape diagram. The number of generations shown can be configured, it is also possible to zoom in and out the view.





# Chapter 7

## Person

Each **person** has a set of values that can be edited:

- **Titles** before and after a full name (e.g., *M.Sc.* or *Ph.D.*)
- **Name** (e.g., *John*)
- **Surname** (e.g., *Swanson*)
- **Birthname** (e.g., *Smith*)
- **Gender** (e.g., *Male*)
- **Date of birth** (e.g., *23. 2. 2000*)
- **Place of birth** (e.g., *Prague*)
- Flag for an **alive** person, otherwise a **date** and **place of death**

### 7.1 Dates

Every **date** value can actually be:

- **Unknown** as a special value for missing information
- Just a **single date** (e.g., *23. 2. 2000*)
- **Range** of two dates (e.g., *From: 23. 2. 2000 to: 25. 2. 2000*)

Each date can also be associated with an optional **note** allowing to hold any information intended.

### 7.2 Tags

**Tag** is a pair of a **name** and its **value**. Each person can be associated with an arbitrary number of such tags. However, they need to be unique. Whenever a new tag name is used for the first time, it becomes globally available within the entire project.



# Chapter 8

## Relations

**Relation** entity allows to bind two particular persons together to represent their real-world relationship. Each relation has these two pieces of information:

- **First person** (e.g., *John Swanson*)
- **Second person** (e.g., *Roy Swanson*)

In order not to be restrained by a hard-wired list of permitted relations, the concept of a **relation template** was introduced.

### 8.1 Relation Templates

Each template has the following characteristics:

- **Title** (e.g., *Fatherhood*)
- **Description** (e.g., *Relation between a father and a child*)
- Name of the **first person** (e.g., *father*)
- Name of the **second person** (e.g., *child*)
- **Generation difference** (e.g., *-1*)
- **Trait** (e.g., *Fatherhood*)

#### 8.1.1 Generation Difference

**Generation difference** describes which person is older and which one is younger. Any number can be possibly provided, but only **-1**, **0**, and **1** usually make sense.

- **-1** means that the first person is older,
- **0** means that both persons are in the same generation
- **1** means that the first person is younger.

### 8.1.2 Trait

**Trait** determines how a particular **relation** is intended to be interpreted. The given list of possibilities shadows the real world.

- **No special trait** (e.g., *Uncle-Nephew*)
- **Motherhood** (e.g., *Mother-Daughter*)
- **Fatherhood** (e.g., *Father-Son*)
- **Partnership** (e.g., *Marriage*)
- **Sibling** (e.g., *Brother-Sister*)

### 8.1.3 Predefined Templates

[Relation](#) templates must be created first before they can be used. While it is possible to introduce specific user-defined templates, several widely used templates were already predefined and are available for import:

- **Fatherhood**
- **Motherhood**
- **Adoptive motherhood**
- **Adoptive fatherhood**
- **Partnership**
- **Marriage**
- **Brotherhood**
- **Sisterhood**
- **Siblings**

## 8.2 Promotion

When a **person** has several **relations** of the same and supported **trait** (e.g., *Father, Mother, Partner, ...*), one of them can be declared as **promoted**, i.e., considered as the main one.

## 8.3 Suggestions

After a new **relation** is created, several **suggestions** for additional plausible relations that can be inferred are automatically offered in order to smooth the entire process.

- **Partnership relation** -> share children between both of them
- **Sibling relation** -> share parents
- **Parent relation** -> children are bound as siblings, partners as parents, and vice versa

## Chapter 9

# Namespace Index

### 9.1 Namespace List

Here is a list of all namespaces with brief descriptions:

|                            |  |    |
|----------------------------|--|----|
| <a href="#">addfile</a>    | Namespace for strings shown in add file dialog . . . . .                               | 25 |
| <a href="#">button</a>     | Namespace for general texts on buttons . . . . .                                       | 25 |
| <a href="#">error</a>      | Namespace for errors and exceptions messages . . . . .                                 | 26 |
| <a href="#">events</a>     | Namespace for strings for default events . . . . .                                     | 28 |
| <a href="#">gender</a>     | Namespace for constants with gender enumeration . . . . .                              | 31 |
| <a href="#">html</a>       | Namespace for html tags and everything with html . . . . .                             | 32 |
| <a href="#">jsonlabel</a>  | Namespace for all JSON labels . . . . .  | 37 |
| <a href="#">parser</a>     | Namespace for all strings representing files and directories in the database . . . . . | 50 |
| <a href="#">person</a>     | Namespace for strings related to a person . . . . .                                    | 52 |
| <a href="#">relation</a>   | Namespace for strings for default relations . . . . .                                  | 53 |
| <a href="#">tagchooser</a> | Namespace for strings shown in tag chooser dialog . . . . .                            | 59 |
| <a href="#">trait</a>      | Namespace for everything with relation traits . . . . .                                | 59 |
| <a href="#">Ui</a>         | Namespace for classes having all its user interface items . . . . .                    | 60 |



## Chapter 10

# Hierarchical Index

### 10.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

|                                  |     |
|----------------------------------|-----|
| AppSettings . . . . .            | 64  |
| Color . . . . .                  | 75  |
| Date . . . . .                   | 76  |
| Event . . . . .                  | 82  |
| EventTemplate . . . . .          | 99  |
| FamilyTree . . . . .             | 108 |
| File . . . . .                   | 137 |
| Parser . . . . .                 | 188 |
| Person . . . . .                 | 197 |
| QDialog                          |     |
| AddFileDialog . . . . .          | 61  |
| ChoosePersonDialog . . . . .     | 72  |
| DateDialog . . . . .             | 80  |
| EventDialog . . . . .            | 89  |
| EventTemplatesDialog . . . . .   | 95  |
| ExportProjectDialog . . . . .    | 105 |
| ExportTemplatesDialog . . . . .  | 107 |
| HelpDialog . . . . .             | 148 |
| HtmlExport . . . . .             | 150 |
| PictureExport . . . . .          | 222 |
| RelationDialog . . . . .         | 232 |
| RelationTemplateDialog . . . . . | 245 |
| SuggestionsDialog . . . . .      | 259 |
| TagChooserDialog . . . . .       | 262 |
| VisualSettings . . . . .         | 270 |
| QGraphicsItem                    |     |
| CellGraphicsItem . . . . .       | 69  |
| PersonsGraphicsItem . . . . .    | 216 |
| QGraphicsView                    |     |
| GraphicsView . . . . .           | 147 |
| QMainWindow                      |     |
| MainWindow . . . . .             | 152 |
| QObject                          |     |
| PersonsGraphicsItem . . . . .    | 216 |
| QTreeWidgetItem                  |     |

|                              |     |
|------------------------------|-----|
| EventRoleItem . . . . .      | 93  |
| FileTreeItem . . . . .       | 141 |
| ProjectItem . . . . .        | 223 |
| RelationItem . . . . .       | 235 |
| Relation . . . . .           | 225 |
| RelationSuggestion . . . . . | 237 |
| RelationTemplate . . . . .   | 240 |
| Settings . . . . .           | 249 |
| VirtualDrive . . . . .       | 264 |
| WrappedDate . . . . .        | 275 |



# Chapter 11

## Class Index

### 11.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

|                                       |  |     |
|---------------------------------------|--|-----|
| <a href="#">AddFileDialog</a>         | Dialog for adding file . . . . .   | 61  |
| <a href="#">AppSettings</a>           | Struct for app settings . . . . .  | 64  |
| <a href="#">CellGraphicsItem</a>      | Cell around all relations . . . . .  | 69  |
| <a href="#">ChoosePersonDialog</a>    | Dialog for choosing person with find window. Parent have to have slot <code>savePerson()</code> . . .  | 72  |
| <a href="#">Color</a>                 | Struct for general color . . . . .   | 75  |
| <a href="#">Date</a>                  | Basic date format. Holding three numbers for year, month and day. If some of the numbers are not defined 0 is present . . . . .                            | 76  |
| <a href="#">DateDialog</a>            | Dialog for editing custom dates . . . . .  | 80  |
| <a href="#">Event</a>                 | Class holding data for a single event . . . . .  | 82  |
| <a href="#">EventDialog</a>           | Dialog showing single event for editing it . . . . .   | 89  |
| <a href="#">EventRoleItem</a>         | Tree widget item for persons in events . . . . .   | 93  |
| <a href="#">EventTemplatesDialog</a>  | Dialog showing all event templates and possibly editing them or creating new ones . . . . .  | 95  |
| <a href="#">EventTemplate</a>         | Template for events. It specifies whether the event includes a date, a place or if more persons are involved and if they have some special roles . . . . . | 99  |
| <a href="#">ExportProjectDialog</a>   | Dialog for exporting project . . . . .   | 105 |
| <a href="#">ExportTemplatesDialog</a> | Dialog for exporting some of the event and relation templates . . . . .  | 107 |
| <a href="#">FamilyTree</a>            | Main class for holding all data of a family tree. Also it acts as a bridge to the core of the application  | 108 |
| <a href="#">File</a>                  | Meta-data about one file (media or note). Its real path is [id]-[filename] . . . . .   | 137 |
| <a href="#">FileTreeItem</a>          | TreeWidget Item for general file . . . . .   | 141 |

|  |  |     |
|--|--|-----|
| <a href="#">GraphicsView</a>           | Class for creating QGraphicsView with using wheel for zoom in and out. And grabbing by mouse moves the view . . . . .                          | 147 |
| <a href="#">HelpDialog</a>             | Dialog with text edit to show help or other texts . . . . .  | 148 |
| <a href="#">HtmlExport</a>             | Dialog for exporting HTML for one person and all persons . . . . .   | 150 |
| <a href="#">MainWindow</a>             | Class representing main window using Qt library . . . . .  | 152 |
| <a href="#">Parser</a>                 | Main class for working with the database . . . . .   | 188 |
| <a href="#">Person</a>                 | One person in the family tree . . . . .  | 197 |
| <a href="#">PersonsGraphicsItem</a>    | Box with persons information . . . . .   | 216 |
| <a href="#">PictureExport</a>          | Dialog for exporting diagram. To choose zoom of the picture . . . . .  | 222 |
| <a href="#">ProjectItem</a>            | This is a tree widget item to be shown on a project view . . . . .   | 223 |
| <a href="#">Relation</a>               | Class holding data for a relationship . . . . .  | 225 |
| <a href="#">RelationDialog</a>         | Dialog showing single relation for editing it . . . . .  | 232 |
| <a href="#">RelationItem</a>           | Class for showing relation suggestion and holding the data . . . . .   | 235 |
| <a href="#">RelationSuggestion</a>     | Struct for relation suggestions . . . . .  | 237 |
| <a href="#">RelationTemplate</a>       | Template for relation. What is the name, description. What is the generation difference. And how are persons in this relation called . . . . . | 240 |
| <a href="#">RelationTemplateDialog</a> | Dialog showing all templates and possibly editing them or creating new ones . . . . .  | 245 |
| <a href="#">Settings</a>               | Class holding all data about the settings of the app and all templates . . . . .   | 249 |
| <a href="#">SuggestionsDialog</a>      | Dialog for relation suggestions . . . . .  | 259 |
| <a href="#">TagChooserDialog</a>       | Dialog for editing or creating tags for a person . . . . .   | 262 |
| <a href="#">VirtualDrive</a>           | Class for virtualization of folder structure for each person . . . . .   | 264 |
| <a href="#">VisualSettings</a>         | Dialog for editing and showing graphic settings . . . . .  | 270 |
| <a href="#">WrappedDate</a>            | Class holding two dates and date description. Always first date is the main one . . . . .  | 275 |

# Chapter 12

## File Index

### 12.1 File List

Here is a list of all files with brief descriptions:

|  |  |     |
|--|--|-----|
| <a href="#">main.cpp</a>                     | The main file for starting the application . . . . .   | 331 |
| core/ <a href="#">config.cpp</a>             | Source code for configuring the app such as special evens and relations . . . . .  | 281 |
| core/ <a href="#">config.h</a>               | Header file for configuration of the app. Also includes templates for events and relations . . . .   | 283 |
| core/ <a href="#">date.cpp</a>               | Source file for date formats . . . . .   | 289 |
| core/ <a href="#">date.h</a>                 | Header file for date formats . . . . .   | 290 |
| core/ <a href="#">family_tree.cpp</a>        | Source file for Family tree . . . . .  | 292 |
| core/ <a href="#">family_tree.h</a>          | Header file for Family tree . . . . .  | 292 |
| core/ <a href="#">family_tree_items.cpp</a>  | Source file for classes closely related to family tree . . . . .   | 295 |
| core/ <a href="#">family_tree_items.h</a>    | Header file for classes closely related to family tree such as <a href="#">Event</a> , <a href="#">Relation</a> and <a href="#">File</a> . . . . . | 296 |
| core/ <a href="#">file_parser.cpp</a>        | Source file for working with predefined database . . . . .   | 299 |
| core/ <a href="#">file_parser.h</a>          | The header file for working with the database and other files . . . . .  | 300 |
| core/ <a href="#">person.cpp</a>             | Source file for all classes closely related to person . . . . .  | 302 |
| core/ <a href="#">person.h</a>               | Header file for all classes closely related to person . . . . .  | 303 |
| core/ <a href="#">strings.h</a>              | Is file containing all English strings . . . . .   | 307 |
| graphics/ <a href="#">dialogs.cpp</a>        | Source file for implementing all dialogs . . . . .   | 318 |
| graphics/ <a href="#">dialogs.h</a>          | This file contains all window dialogs that will be shown. These classes are relatively small but there are many of them . . . . .                  | 318 |
| graphics/ <a href="#">graphics_items.cpp</a> | This files contains implementation of all graphics items, that are inherited from Qt framework .   | 323 |

|  |     |
|--|-----|
| graphics/ <a href="#">graphics_items.h</a>   |     |
| This files contains all graphics items, that are inherited from Qt framework                         | 324 |
| graphics/ <a href="#">mainwindow.cpp</a>   |     |
| Implementation of all standard methods. For slots see <a href="#">mainwindow_slots.cpp</a>           | 327 |
| graphics/ <a href="#">mainwindow.h</a>   |     |
| This header file contains everything around the main window  | 328 |
| graphics/ <a href="#">mainwindow_slots.cpp</a>   |     |
| Implementation of all slots of the main window. For other methods see <a href="#">mainwindow.cpp</a> | 331 |

## Chapter 13

# Namespace Documentation

### 13.1 addfile Namespace Reference

Namespace for strings shown in add file dialog.

#### Variables

- `const std::string NONE = "None"`  
*String for no extension.*

#### 13.1.1 Detailed Description

Namespace for strings shown in add file dialog.

#### 13.1.2 Variable Documentation

##### 13.1.2.1 NONE

```
const std::string addfile::NONE = "None"
```

String for no extension.

### 13.2 button Namespace Reference

Namespace for general texts on buttons.

## Variables

- `const std::string ADD_NEW = "Add new"`  
*Text for adding new element.*
- `const std::string CANCEL = "Cancel"`  
*Text for cancelling.*
- `const std::string REMOVE = "Remove"`  
*text for removing element.*

### 13.2.1 Detailed Description

Namespace for general texts on buttons.

### 13.2.2 Variable Documentation

#### 13.2.2.1 ADD\_NEW

```
const std::string button::ADD_NEW = "Add new"
```

Text for adding new element.

#### 13.2.2.2 CANCEL

```
const std::string button::CANCEL = "Cancel"
```

Text for cancelling.

#### 13.2.2.3 REMOVE

```
const std::string button::REMOVE = "Remove"
```

text for removing element.

## 13.3 error Namespace Reference

Namespace for errors and exceptions messages.

## Variables

- `const std::string UNKNOWN_EXCEP_STR = "Unknown exception catch."`  
*String for catching unknown exception.*
- `const std::string EXCEPTION_STR = "Exception: "`  
*String for adding before exception description.*
- `const std::string CORRUPTED_DATABASE = "Given database is corrupted: "`  
*String shown when database is corrupted.*
- `const std::string EMPTY_BOXES = "All boxes must not be empty."`  
*String for showing that some boxes were left empty.*
- `const std::string NO_CHOSEN_FILE = "No file was chosen."`  
*No file chosen and empty box.*
- `const std::string NO_ROLE = "Given role 'no role' cannot be used. It would be considered as no special role."`  
*When person tries to add "no role" role, which is allocated for no roles.*
- `const std::string FORBIDDEN_CHARS = "Text cannot contain forbidden characters. &lt; &gt; \\"`  
*When in any box where inserted forbidden characters.*

### 13.3.1 Detailed Description

Namespace for errors and exceptions messages.

### 13.3.2 Variable Documentation

#### 13.3.2.1 CORRUPTED\_DATABASE

```
const std::string error::CORRUPTED_DATABASE = "Given database is corrupted: "
```

String shown when database is corrupted.

#### 13.3.2.2 EMPTY\_BOXES

```
const std::string error::EMPTY_BOXES = "All boxes must not be empty."
```

String for showing that some boxes were left empty.

#### 13.3.2.3 EXCEPTION\_STR

```
const std::string error::EXCEPTION_STR = "Exception: "
```

String for adding before exception description.

#### 13.3.2.4 FORBIDDEN\_CHARS

```
const std::string error::FORBIDDEN_CHARS = "Text cannot contain forbidden characters. &lt; &gt; \\"
```

When in any box where inserted forbidden characters.

### 13.3.2.5 NO\_CHOSEN\_FILE

```
const std::string error::NO_CHOSEN_FILE = "No file was chosen."
```

No file chosen and empty box.

### 13.3.2.6 NO\_ROLE

```
const std::string error::NO_ROLE = "Given role 'no role' cannot be used. It would be considered as no special role."
```

When person tries to add "no role" role, which is allocated for no roles.

### 13.3.2.7 UNKNOWN\_EXCEP\_STR

```
const std::string error::UNKNOWN_EXCEP_STR = "Unknown exception catch."
```

String for catching unknown exception.

## 13.4 events Namespace Reference

Namespace for strings for default events.

### Variables

- const std::string [WEDDING](#) = "Wedding"  
*String for wedding.*
- const std::string [DESC\\_WEDDING](#) = "A social event mainly between soon to be wife and husband."  
*String for description of wedding.*
- const std::string [GROOM](#) = "Groom"  
*String for groom.*
- const std::string [BRIDE](#) = "Bride"  
*String for bride.*
- const std::string [BEST\\_MAN](#) = "Best man"  
*String for best man.*
- const std::string [MAID\\_OF\\_HONOR](#) = "Maid of Honor"  
*String for maid of honor.*
- const std::string [DIVORCE](#) = "Divorce"  
*String for divorce.*
- const std::string [DESC\\_DIVORCE](#) = "Divorce is the process of terminating a marriage or marital union."  
*String for divorce description.*
- const std::string [FORMER\\_HUSBAND](#) = "Former husband"  
*String for former husband.*
- const std::string [FORMER\\_WIFE](#) = "Former wife"  
*String for former wife.*
- const std::string [FUNERAL](#) = "Funeral"  
*String for funeral.*



- `const std::string DESC_FUNERAL = "A funeral is a ceremony connected with the final disposition of a corpse, such as a burial or cremation, with the attendant observances."`  
*String for funeral description.*
- `const std::string DEAD_PERSON = "Dead person"`  
*String for dead person.*
- `const std::string GRADUATION = "Graduation"`  
*String for graduation.*
- `const std::string DESC_GRADUATION = "Graduation is the awarding of a diploma to a student by an educational institution."`  
*String for graduation description.*
- `const std::string GRADUATE = "Graduate"`  
*String for graduate.*
- `const std::string NO_ROLE = "no role"`  
*String for no role in event.*

### 13.4.1 Detailed Description

Namespace for strings for default events.

### 13.4.2 Variable Documentation

#### 13.4.2.1 BEST\_MAN

```
const std::string events::BEST_MAN = "Best man"
```

String for best man.

#### 13.4.2.2 BRIDE

```
const std::string events::BRIDE = "Bride"
```

String for bride.

#### 13.4.2.3 DEAD\_PERSON

```
const std::string events::DEAD_PERSON = "Dead person"
```

String for dead person.

#### 13.4.2.4 DESC\_DIVORCE

```
const std::string events::DESC_DIVORCE = "Divorce is the process of terminating a marriage or marital union."
```

String for divorce description.

#### 13.4.2.5 DESC\_FUNERAL

```
const std::string events::DESC_FUNERAL = "A funeral is a ceremony connected with the final disposition of a corpse, such as a burial or cremation, with the attendant observances."
```

String for funeral description.

#### 13.4.2.6 DESC\_GRADUATION

```
const std::string events::DESC_GRADUATION = "Graduation is the awarding of a diploma to a student by an educational institution."
```

String for graduation description.

#### 13.4.2.7 DESC\_WEDDING

```
const std::string events::DESC_WEDDING = "A social event mainly between soon to be wife and husband."
```

String for description of wedding.

#### 13.4.2.8 DIVORCE

```
const std::string events::DIVORCE = "Divorce"
```

String for divorce.

#### 13.4.2.9 FORMER\_HUSBAND

```
const std::string events::FORMER_HUSBAND = "Former husband"
```

String for former husband.

#### 13.4.2.10 FORMER\_WIFE

```
const std::string events::FORMER_WIFE = "Former wife"
```

String for former wife.

#### 13.4.2.11 FUNERAL

```
const std::string events::FUNERAL = "Funeral"
```

String for funeral.

#### 13.4.2.12 GRADUATE

```
const std::string events::GRADUATE = "Graduate"
```

String for graduate.

#### 13.4.2.13 GRADUATION

```
const std::string events::GRADUATION = "Graduation"
```

String for graduation.

#### 13.4.2.14 GROOM

```
const std::string events::GROOM = "Groom"
```

String for groom.

#### 13.4.2.15 MAID\_OF\_HONOR

```
const std::string events::MAID_OF_HONOR = "Maid of Honor"
```

String for maid of honor.

#### 13.4.2.16 NO\_ROLE

```
const std::string events::NO_ROLE = "no role"
```

String for no role in event.

#### 13.4.2.17 WEDDING

```
const std::string events::WEDDING = "Wedding"
```

String for wedding.

## 13.5 gender Namespace Reference

Namespace for constants with gender enumeration.

### Functions

- `std::string str (Gender g)`

## Variables

- constexpr `Gender AllGenders [] = {Other, Male, Female}`  
*All genders in a list.*

### 13.5.1 Detailed Description

Namespace for constants with gender enumeration.

### 13.5.2 Function Documentation

#### 13.5.2.1 `str()`

```
std::string gender::str (
    Gender g )
```

Write the gender as a string.

#### Parameters

|          |               |
|----------|---------------|
| <i>g</i> | Given Gender. |
|----------|---------------|

#### Returns

Newly constructed string representing the gender.

### 13.5.3 Variable Documentation

#### 13.5.3.1 `AllGenders`

```
constexpr Gender gender::AllGenders[] = {Other, Male, Female} [constexpr]
```

All genders in a list.

## 13.6 html Namespace Reference

Namespace for html tags and everything with html.

## Variables

- const std::string `RESOURCES` = "resources"  
*Where will resources be saved.*
- const std::string `PREFIX_CLOSE` = "/close-"  
*Prefix for close family diagram.*
- const std::string `PREFIX_FAMILY_TREE` = "/tree-"

- Prefix for family tree diagram.*

  - const std::string **CUSTOM\_CSS** = "styles.css"

*Custom CSS file name.*
- const std::string **GENERAL\_INFO** = "General information"

*General information.*
- const std::string **BIRTH\_DATE** = "Birth date: "

*Birth date.*
- const std::string **BIRTH\_PLACE** = "Birth place: "

*Birth place.*
- const std::string **DEATH\_DATE** = "Date of death: "

*Date of death.*
- const std::string **DEATH\_PLACE** = "Place of death: "

*Place of death.*
- const std::string **GENDER** = "Gender: "

*Gender.*
- const std::string **CUSTOM\_TAGS** = "Custom tags:"

*Custom tags.*
- const std::string **DESCRIPTION** = "Description:"

*Description.*
- const std::string **NOTE** = "Note:"

*Note.*
- const std::string **DATE** = "Date: "

*Date.*
- const std::string **PLACE** = "Place: "

*Place.*
- const std::string **PERSONS** = "Persons:"

*Persons.*
- const std::string **ALL\_PERSONS** = "All persons"

*All persons.*
- const std::string **EVENTS** = "Events"

*Events.*
- const std::string **FAMILY\_TREE** = "Family tree"

*Family tree.*
- const std::string **FAMILY\_TREE\_DIAGRAM** = "Family tree diagram"

*Family tree diagram.*
- const std::string **RELATIONS** = "Relations"

*Relations.*
- const std::string **CLOSE\_FAMILY** = "Close family"

*Close family.*
- const std::string **CLOSE\_FAMILY\_DIAGRAM** = "Close family diagram"

*Close family diagram.*

### 13.6.1 Detailed Description

Namespace for html tags and everything with html.

## 13.6.2 Variable Documentation

### 13.6.2.1 ALL\_PERSONS

```
const std::string html::ALL_PERSONS = "All persons"
```

All persons.

### 13.6.2.2 BIRTH\_DATE

```
const std::string html::BIRTH_DATE = "Birth date: "
```

Birth date.

### 13.6.2.3 BIRTH\_PLACE

```
const std::string html::BIRTH_PLACE = "Birth place: "
```

Birth place.

### 13.6.2.4 CLOSE\_FAMILY

```
const std::string html::CLOSE_FAMILY = "Close family"
```

Close family.

### 13.6.2.5 CLOSE\_FAMILY\_DIAGRAM

```
const std::string html::CLOSE_FAMILY_DIAGRAM = "Close family diagram"
```

Close family diagram.

### 13.6.2.6 CUSTOM\_CSS

```
const std::string html::CUSTOM_CSS = "styles.css"
```

Custom CSS file name.

### 13.6.2.7 CUSTOM\_TAGS

```
const std::string html::CUSTOM_TAGS = "Custom tags:"
```

Custom tags.

### 13.6.2.8 DATE

```
const std::string html::DATE = "Date:  "
```

[Date](#).

### 13.6.2.9 DEATH\_DATE

```
const std::string html::DEATH_DATE = "Date of death:  "
```

[Date](#) of death.

### 13.6.2.10 DEATH\_PLACE

```
const std::string html::DEATH_PLACE = "Place of death:  "
```

Place of death.

### 13.6.2.11 DESCRIPTION

```
const std::string html::DESCRIPTION = "Description:"
```

Description.

### 13.6.2.12 EVENTS

```
const std::string html::EVENTS = "Events"
```

Events.

### 13.6.2.13 FAMILY\_TREE

```
const std::string html::FAMILY_TREE = "Family tree"
```

Family tree.

### 13.6.2.14 FAMILY\_TREE\_DIAGRAM

```
const std::string html::FAMILY_TREE_DIAGRAM = "Family tree diagram"
```

Family tree diagram.

### 13.6.2.15 GENDER

```
const std::string html::GENDER = "Gender:  "
```

Gender.

#### 13.6.2.16 GENERAL\_INFO

```
const std::string html::GENERAL_INFO = "General information"
```

General information.

#### 13.6.2.17 NOTE

```
const std::string html::NOTE = "Note:"
```

Note.

#### 13.6.2.18 PERSONS

```
const std::string html::PERSONS = "Persons:"
```

Persons.

#### 13.6.2.19 PLACE

```
const std::string html::PLACE = "Place:  "
```

Place.

#### 13.6.2.20 PREFIX\_CLOSE

```
const std::string html::PREFIX_CLOSE = "/close-"
```

Prefix for close family diagram.

#### 13.6.2.21 PREFIX\_FAMILY\_TREE

```
const std::string html::PREFIX_FAMILY_TREE = "/tree-"
```

Prefix for family tree diagram.

#### 13.6.2.22 RELATIONS

```
const std::string html::RELATIONS = "Relations"
```

Relations.

#### 13.6.2.23 RESOURCES

```
const std::string html::RESOURCES = "resources"
```

Where will resources be saved.



## 13.7 jsonlabel Namespace Reference

Namespace for all JSON labels.

### Variables

- const std::string **TITLE** = "title"  
*JSON Label for title.*
- const std::string **ID** = "id"  
*JSON Label for id.*
- const std::string **DESCRIPTION** = "description"  
*JSON Label for description.*
- const std::string **CONTAINS\_DATE** = "contains date"  
*JSON Label for if the event contains a date or not.*
- const std::string **MORE\_PEOPLE** = "more people"  
*JSON Label for if the event contains more people.*
- const std::string **SPECIAL\_ROLES** = "special roles"  
*JSON Label for if the event has special roles.*
- const std::string **NAME** = "name"  
*JSON Label for name.*
- const std::string **FIRST\_NAME** = "first name"  
*JSON Label for first name.*
- const std::string **SECOND\_NAME** = "second name"  
*JSON Label for second name.*
- const std::string **TRAIT** = "trait"  
*JSON Label for trait.*
- const std::string **GEN\_DIFF** = "generation difference"  
*JSON Label for generation difference.*
- const std::string **REL\_TEMPL** = "relation templates"  
*JSON Label for relations templates.*
- const std::string **EVE\_TEMPL** = "event templates"  
*JSON Label for event templates.*
- const std::string **TYPE** = "type"  
*JSON Label for type.*
- const std::string **DATE** = "date"  
*JSON Label for date.*
- const std::string **LAST\_DATE** = "last date"  
*JSON Label for last date.*
- const std::string **TEXT** = "text"  
*JSON Label for text.*
- const std::string **TAGS** = "tags"  
*JSON Label for tags.*
- const std::string **TAG** = "tag"  
*JSON Label for single tag.*
- const std::string **RELATIONS** = "relations"  
*JSON Label for relations.*
- const std::string **EVENTS** = "events"  
*JSON Label for events.*
- const std::string **SURNAME** = "surname"  
*JSON Label for surname.*

- `const std::string MAIDEN_NAME = "maiden name"`  
*JSON Label for maiden name.*
- `const std::string VALUE = "value"`  
*JSON Label for value.*
- `const std::string BIRTH = "birth"`  
*JSON Label for birth.*
- `const std::string DEATH = "death"`  
*JSON Label for death.*
- `const std::string VIRTUAL_DRIVE = "virtual drive"`  
*JSON Label for virtual drive.*
- `const std::string MEDIA = "media"`  
*JSON Label for media.*
- `const std::string NOTES = "notes"`  
*JSON Label for notes.*
- `const std::string FILES = "files"`  
*JSON Label for files.*
- `const std::string PERSONS = "persons"`  
*JSON Label for persons.*
- `const std::string TEMPLATE = "template"`  
*JSON Label for for template.*
- `const std::string TEMPLATES = "templates"`  
*JSON Label for for templates.*
- `const std::string FIRST_PERSON = "first person"`  
*JSON Label for first person.*
- `const std::string SECOND_PERSON = "second person"`  
*JSON Label for second person.*
- `const std::string PLACE = "place"`  
*JSON Label for place.*
- `const std::string ROLE = "role"`  
*JSON Label for role.*
- `const std::string PERSON = "person"`  
*JSON Label for person.*
- `const std::string GENDER = "gender"`  
*JSON Label for gender.*
- `const std::string LIVES = "lives"`  
*JSON Label for if the person lives or not.*
- `const std::string FRONT_TITLE = "title in front"`  
*JSON Label for title in front of a name.*
- `const std::string AFTER_TITLE = "title after"`  
*JSON Label for title after the name.*
- `const std::string FATHER = "father"`  
*JSON Label for father.*
- `const std::string MOTHER = "mother"`  
*JSON Label for mother.*
- `const std::string PARTNER = "partner"`  
*JSON Label for partner.*
- `const std::string MOVEABLE_TABS = "moveable tabs"`  
*JSON Label for setting moveable tabs.*
- `const std::string TAB_POSITION = "tab position"`  
*JSON Label for tab position.*
- `const std::string GLOBAL_MAIN_PERSON = "global main person"`

- JSON Label for global main person.*

  - const std::string **CONTAINS\_PLACE** = "contains place"

*JSON Label if an event template contains a place.*
- const std::string **SHOW\_GENERAL\_TAB** = "show general tab"

*JSON Label for app settings if general tab should be visible.*
- const std::string **SHOW\_EVENT\_TAB** = "show event tab"

*JSON Label for app settings if event tab should be visible.*
- const std::string **SHOW\_FILE\_TAB** = "show file tab"

*JSON Label for app settings if file tab should be visible.*
- const std::string **SHOW\_MEDIA\_TAB** = "show media tab"

*JSON Label for app settings if media tab should be visible.*
- const std::string **SHOW\_NOTE\_TAB** = "show note tab"

*JSON Label for app settings if note tab should be visible.*
- const std::string **SHOW\_RELATION\_TAB** = "show relation tab"

*JSON Label for app settings if relation tab should be visible.*
- const std::string **APP\_SETTINGS** = "app settings"

*JSON Label for app settings.*
- const std::string **DRIVES** = "drives"

*JSON Label for drives.*
- const std::string **EXPORT** = "export"

*JSON Label for export.*
- const std::string **RED** = "red"

*JSON Label for red.*
- const std::string **BLUE** = "blue"

*JSON Label for blue.*
- const std::string **GREEN** = "green"

*JSON Label for green.*
- const std::string **FONT\_FAMILY** = "font family"

*JSON Label for font family.*
- const std::string **FONT\_SIZE** = "font size"

*JSON Label for font size.*
- const std::string **HIGHLIGHTED\_COLOR** = "highlighted color"

*JSON Label highlighted color.*
- const std::string **PROBANT\_COLOR** = "proband color"

*JSON Label for proband color.*
- const std::string **PROMOTED\_COLOR** = "promoted color"

*JSON Label promoted color.*
- const std::string **STANDARD\_COLOR** = "standard color"

*JSON Label standard color.*
- const std::string **X\_SIZE** = "x size"

*JSON Label for size in the x axis.*
- const std::string **Y\_SIZE** = "y size"

*JSON Label for size in the y axis.*
- const std::string **GEN\_SIZE\_UP** = "number of generations up"

*JSON Label for how many generations are being shown upwards.*
- const std::string **GEN\_SIZE\_DOWN** = "number of generations down"

*JSON Label for how many generations are being shown downwards.*
- const std::string **DIR** = "dir"

*JSON Label for directory.*
- const std::string **ROOT** = "root"

*JSON Label for root.*

- `const std::string HELP = "help"`  
*JSON Label for help.*
- `const std::string SPLITTER_POSITION_ONE = "splitter position one"`  
*JSON Label for splitter position one.*
- `const std::string SPLITTER_POSITION_TWO = "splitter position two"`  
*JSON Label for splitter position two.*
- `const std::string TEXT_COLOR = "text color"`  
*JSON Label for the color of text.*
- `const std::string LINE_COLOR = "line color"`  
*JSON Label for the line color.*
- `const std::string LINE_WIDTH = "line width"`  
*JSON Label for line width.*
- `const std::string BORDER_RADIUS = "radius"`  
*JSON Label border radius.*

### 13.7.1 Detailed Description

Namespace for all JSON labels.

### 13.7.2 Variable Documentation

#### 13.7.2.1 AFTER\_TITLE

```
const std::string jsonlabel::AFTER_TITLE = "title after"
```

JSON Label for title after the name.

#### 13.7.2.2 APP\_SETTINGS

```
const std::string jsonlabel::APP_SETTINGS = "app settings"
```

JSON Label for app settings.

#### 13.7.2.3 BIRTH

```
const std::string jsonlabel::BIRTH = "birth"
```

JSON Label for birth.

#### 13.7.2.4 BLUE

```
const std::string jsonlabel::BLUE = "blue"
```

JSON Label for blue.

### 13.7.2.5 BORDER\_RADIUS

```
const std::string jsonlabel::BORDER_RADIUS = "radius"
```

JSON Label border radius.

### 13.7.2.6 CONTAINS\_DATE

```
const std::string jsonlabel::CONTAINS_DATE = "contains date"
```

JSON Label for if the event contains a date or not.

### 13.7.2.7 CONTAINS\_PLACE

```
const std::string jsonlabel::CONTAINS_PLACE = "contains place"
```

JSON Label if an event template contains a place.

### 13.7.2.8 DATE

```
const std::string jsonlabel::DATE = "date"
```

JSON Label for date.

### 13.7.2.9 DEATH

```
const std::string jsonlabel::DEATH = "death"
```

JSON Label for death.

### 13.7.2.10 DESCRIPTION

```
const std::string jsonlabel::DESCRIPTION = "description"
```

JSON Label for description.

### 13.7.2.11 DIR

```
const std::string jsonlabel::DIR = "dir"
```

JSON Label for directory.

### 13.7.2.12 DRIVES

```
const std::string jsonlabel::DRIVES = "drives"
```

JSON Label for drives.

### 13.7.2.13 EVE\_TEMPL

```
const std::string jsonlabel::EVE_TEMPL = "event templates"
```

JSON Label for event templates.

### 13.7.2.14 EVENTS

```
const std::string jsonlabel::EVENTS = "events"
```

JSON Label for events.

### 13.7.2.15 EXPORT

```
const std::string jsonlabel::EXPORT = "export"
```

JSON Label for export.

### 13.7.2.16 FATHER

```
const std::string jsonlabel::FATHER = "father"
```

JSON Label for father.

### 13.7.2.17 FILES

```
const std::string jsonlabel::FILES = "files"
```

JSON Label for files.

### 13.7.2.18 FIRST\_NAME

```
const std::string jsonlabel::FIRST_NAME = "first name"
```

JSON Label for first name.

### 13.7.2.19 FIRST\_PERSON

```
const std::string jsonlabel::FIRST_PERSON = "first person"
```

JSON Label for first person.

### 13.7.2.20 FONT\_FAMILY

```
const std::string jsonlabel::FONT_FAMILY = "font family"
```

JSON Label for font family.

#### 13.7.2.21 FONT\_SIZE

```
const std::string jsonlabel::FONT_SIZE = "font size"
```

JSON Label for font size.

#### 13.7.2.22 FRONT\_TITLE

```
const std::string jsonlabel::FRONT_TITLE = "title in front"
```

JSON Label for title in front of a name.

#### 13.7.2.23 GEN\_DIFF

```
const std::string jsonlabel::GEN_DIFF = "generation difference"
```

JSON Label for generation difference.

#### 13.7.2.24 GEN\_SIZE\_DOWN

```
const std::string jsonlabel::GEN_SIZE_DOWN = "number of generations down"
```

JSON Label for how many generations are being shown downwards.

#### 13.7.2.25 GEN\_SIZE\_UP

```
const std::string jsonlabel::GEN_SIZE_UP = "number of generations up"
```

JSON Label for how many generations are being shown upwards.

#### 13.7.2.26 GENDER

```
const std::string jsonlabel::GENDER = "gender"
```

JSON Label for gender.

#### 13.7.2.27 GLOBAL\_MAIN\_PERSON

```
const std::string jsonlabel::GLOBAL_MAIN_PERSON = "global main person"
```

JSON Label for global main person.

#### 13.7.2.28 GREEN

```
const std::string jsonlabel::GREEN = "green"
```

JSON Label for green.

#### 13.7.2.29 HELP

```
const std::string jsonlabel::HELP = "help"
```

JSON Label for help.

#### 13.7.2.30 HIGHLIGHTED\_COLOR

```
const std::string jsonlabel::HIGHLIGHTED_COLOR = "highlighted color"
```

JSON Label highlighted color.

#### 13.7.2.31 ID

```
const std::string jsonlabel::ID = "id"
```

JSON Label for id.

#### 13.7.2.32 LAST\_DATE

```
const std::string jsonlabel::LAST_DATE = "last date"
```

JSON Label for last date.

#### 13.7.2.33 LINE\_COLOR

```
const std::string jsonlabel::LINE_COLOR = "line color"
```

JSON Label for the line color.

#### 13.7.2.34 LINE\_WIDTH

```
const std::string jsonlabel::LINE_WIDTH = "line width"
```

JSON Label for line width.

#### 13.7.2.35 LIVES

```
const std::string jsonlabel::LIVES = "lives"
```

JSON Label for if the person lives or not.

#### 13.7.2.36 MAIDEN\_NAME

```
const std::string jsonlabel::MAIDEN_NAME = "maiden name"
```

JSON Label for maiden name.



### 13.7.2.37 MEDIA

```
const std::string jsonlabel::MEDIA = "media"
```

JSON Label for media.

### 13.7.2.38 MORE\_PEOPLE

```
const std::string jsonlabel::MORE_PEOPLE = "more people"
```

JSON Label for if the event contains more people.

### 13.7.2.39 MOTHER

```
const std::string jsonlabel::MOTHER = "mother"
```

JSON Label for mother.

### 13.7.2.40 MOVEABLE\_TABS

```
const std::string jsonlabel::MOVEABLE_TABS = "moveable tabs"
```

JSON Label for setting moveable tabs.

### 13.7.2.41 NAME

```
const std::string jsonlabel::NAME = "name"
```

JSON Label for name.

### 13.7.2.42 NOTES

```
const std::string jsonlabel::NOTES = "notes"
```

JSON Label for notes.

### 13.7.2.43 PARTNER

```
const std::string jsonlabel::PARTNER = "partner"
```

JSON Label for partner.

### 13.7.2.44 PERSON

```
const std::string jsonlabel::PERSON = "person"
```

JSON Label for person.

#### 13.7.2.45 PERSONS

```
const std::string jsonlabel::PERSONS = "persons"
```

JSON Label for persons.

#### 13.7.2.46 PLACE

```
const std::string jsonlabel::PLACE = "place"
```

JSON Label for place.

#### 13.7.2.47 PROBANT\_COLOR

```
const std::string jsonlabel::PROBANT_COLOR = "proband color"
```

JSON Label for proband color.

#### 13.7.2.48 PROMOTED\_COLOR

```
const std::string jsonlabel::PROMOTED_COLOR = "promoted color"
```

JSON Label promoted color.

#### 13.7.2.49 RED

```
const std::string jsonlabel::RED = "red"
```

JSON Label for red.

#### 13.7.2.50 REL\_TEMPL

```
const std::string jsonlabel::REL_TEMPL = "relation templates"
```

JSON Label for relations templates.

#### 13.7.2.51 RELATIONS

```
const std::string jsonlabel::RELATIONS = "relations"
```

JSON Label for relations.

#### 13.7.2.52 ROLE

```
const std::string jsonlabel::ROLE = "role"
```

JSON Label for role.

### 13.7.2.53 ROOT

```
const std::string jsonlabel::ROOT = "root"
```

JSON Label for root.

### 13.7.2.54 SECOND\_NAME

```
const std::string jsonlabel::SECOND_NAME = "second name"
```

JSON Label for second name.

### 13.7.2.55 SECOND\_PERSON

```
const std::string jsonlabel::SECOND_PERSON = "second person"
```

JSON Label for second person.

### 13.7.2.56 SHOW\_EVENT\_TAB

```
const std::string jsonlabel::SHOW_EVENT_TAB = "show event tab"
```

JSON Label for app settings if event tab should be visible.

### 13.7.2.57 SHOW\_FILE\_TAB

```
const std::string jsonlabel::SHOW_FILE_TAB = "show file tab"
```

JSON Label for app settings if file tab should be visible.

### 13.7.2.58 SHOW\_GENERAL\_TAB

```
const std::string jsonlabel::SHOW_GENERAL_TAB = "show general tab"
```

JSON Label for app settings if general tab should be visible.

### 13.7.2.59 SHOW\_MEDIA\_TAB

```
const std::string jsonlabel::SHOW_MEDIA_TAB = "show media tab"
```

JSON Label for app settings if media tab should be visible.

### 13.7.2.60 SHOW\_NOTE\_TAB

```
const std::string jsonlabel::SHOW_NOTE_TAB = "show note tab"
```

JSON Label for app settings if note tab should be visible.

#### 13.7.2.61 SHOW\_RELATION\_TAB

```
const std::string jsonlabel::SHOW_RELATION_TAB = "show relation tab"
```

JSON Label for app settings if relation tab should be visible.

#### 13.7.2.62 SPECIAL\_ROLES

```
const std::string jsonlabel::SPECIAL_ROLES = "special roles"
```

JSON Label for if the event has special roles.

#### 13.7.2.63 SPLITTER\_POSITION\_ONE

```
const std::string jsonlabel::SPLITTER_POSITION_ONE = "splitter position one"
```

JSON Label for splitter position one.

#### 13.7.2.64 SPLITTER\_POSITION\_TWO

```
const std::string jsonlabel::SPLITTER_POSITION_TWO = "splitter position two"
```

JSON Label for splitter position two.

#### 13.7.2.65 STANDARD\_COLOR

```
const std::string jsonlabel::STANDARD_COLOR = "standard color"
```

JSON Label standard color.

#### 13.7.2.66 SURNAME

```
const std::string jsonlabel::SURNAME = "surname"
```

JSON Label for surname.

#### 13.7.2.67 TAB\_POSITION

```
const std::string jsonlabel::TAB_POSITION = "tab position"
```

JSON Label for tab position.

#### 13.7.2.68 TAG

```
const std::string jsonlabel::TAG = "tag"
```

JSON Label for single tag.

### 13.7.2.69 TAGS

```
const std::string jsonlabel::TAGS = "tags"
```

JSON Label for tags.

### 13.7.2.70 TEMPLATE

```
const std::string jsonlabel::TEMPLATE = "template"
```

JSON Label for for template.

### 13.7.2.71 TEMPLATES

```
const std::string jsonlabel::TEMPLATES = "templates"
```

JSON Label for for templates.

### 13.7.2.72 TEXT

```
const std::string jsonlabel::TEXT = "text"
```

JSON Label for text.

### 13.7.2.73 TEXT\_COLOR

```
const std::string jsonlabel::TEXT_COLOR = "text color"
```

JSON Label for the color of text.

### 13.7.2.74 TITLE

```
const std::string jsonlabel::TITLE = "title"
```

JSON Label for title.

### 13.7.2.75 TRAIT

```
const std::string jsonlabel::TRAIT = "trait"
```

JSON Label for trait.

### 13.7.2.76 TYPE

```
const std::string jsonlabel::TYPE = "type"
```

JSON Label for type.

### 13.7.2.77 VALUE

```
const std::string jsonlabel::VALUE = "value"
```

JSON Label for value.

### 13.7.2.78 VIRTUAL\_DRIVE

```
const std::string jsonlabel::VIRTUAL_DRIVE = "virtual drive"
```

JSON Label for virtual drive.

### 13.7.2.79 X\_SIZE

```
const std::string jsonlabel::X_SIZE = "x size"
```

JSON Label for size in the x axis.

### 13.7.2.80 Y\_SIZE

```
const std::string jsonlabel::Y_SIZE = "y size"
```

JSON Label for size in the y axis.

## 13.8 parser Namespace Reference

Namespace for all strings representing files and directories in the database.

### Variables

- const std::string [FILES\\_DIR](#) = "Files"  
*Directory for all documents and files in general.*
- const std::string [JSON\\_CONFIG](#) = "Config.json"  
*JSON file-name for configuration and templates.*
- const std::string [JSON\\_EVENTS](#) = "Events.json"  
*JSON file-name for events.*
- const std::string [JSON\\_FILES](#) = "Files.json"  
*JSON file-name for files meta-data.*
- const std::string [JSON\\_MEDIA](#) = "Media.json"  
*JSON file-name for media meta-data.*
- const std::string [JSON\\_NOTES](#) = "Notes.json"  
*JSON file-name for notes meta-data.*
- const std::string [JSON\\_PERSONS](#) = "Persons.json"  
*JSON file-name for persons.*
- const std::string [JSON\\_RELATIONS](#) = "Relations.json"  
*JSON file-name for relations.*
- const std::string [MEDIA\\_DIR](#) = "Media"  
*Directory for all pictures and other media.*
- const std::string [NOTES\\_DIR](#) = "Notes"  
*Directory for all notes.*
- const std::string [ERROR\\_LOG](#) = ".error.log"  
*File for writing error log.*

### 13.8.1 Detailed Description

Namespace for all strings representing files and directories in the database.

### 13.8.2 Variable Documentation

#### 13.8.2.1 ERROR\_LOG

```
const std::string parser::ERROR_LOG = ".error.log"
```

[File](#) for writing error log.

#### 13.8.2.2 FILES\_DIR

```
const std::string parser::FILES_DIR = "Files"
```

Directory for all documents and files in general.

#### 13.8.2.3 JSON\_CONFIG

```
const std::string parser::JSON_CONFIG = "Config.json"
```

JSON file-name for configuration and templates.

#### 13.8.2.4 JSON\_EVENTS

```
const std::string parser::JSON_EVENTS = "Events.json"
```

JSON file-name for events.

#### 13.8.2.5 JSON\_FILES

```
const std::string parser::JSON_FILES = "Files.json"
```

JSON file-name for files meta-data.

#### 13.8.2.6 JSON\_MEDIA

```
const std::string parser::JSON_MEDIA = "Media.json"
```

JSON file-name for media meta-data.

### 13.8.2.7 JSON\_NOTES

```
const std::string parser::JSON_NOTES = "Notes.json"
```

JSON file-name for notes meta-data.

### 13.8.2.8 JSON\_PERSONS

```
const std::string parser::JSON_PERSONS = "Persons.json"
```

JSON file-name for persons.

### 13.8.2.9 JSON\_RELATIONS

```
const std::string parser::JSON_RELATIONS = "Relations.json"
```

JSON file-name for relations.

### 13.8.2.10 MEDIA\_DIR

```
const std::string parser::MEDIA_DIR = "Media"
```

Directory for all pictures and other media.

### 13.8.2.11 NOTES\_DIR

```
const std::string parser::NOTES_DIR = "Notes"
```

Directory for all notes.

## 13.9 person Namespace Reference

Namespace for strings related to a person.

### Variables

- const std::string [UNKNOWN\\_STR](#) = "Unknown"  
*String for unknown tags.*
- const std::string [SURNAME\\_STR](#) = "Surname"  
*String for default surname.*
- const std::string [NAME\\_STR](#) = "Name"  
*String for default name.*
- const std::string [ROOT](#) = "root"  
*String for default root directories.*
- const std::string [DATE\\_FROM](#) = "From: "  
*String for date interval from.*
- const std::string [DATE\\_TO](#) = " to: "  
*String for date interval to.*



### 13.9.1 Detailed Description

Namespace for strings related to a person.

### 13.9.2 Variable Documentation

#### 13.9.2.1 DATE\_FROM

```
const std::string person::DATE_FROM = "From: "
```

String for date interval from.

#### 13.9.2.2 DATE\_TO

```
const std::string person::DATE_TO = " to: "
```

String for date interval to.

#### 13.9.2.3 NAME\_STR

```
const std::string person::NAME_STR = "Name"
```

String for default name.

#### 13.9.2.4 ROOT

```
const std::string person::ROOT = "root"
```

String for default root directories.

#### 13.9.2.5 SURNAME\_STR

```
const std::string person::SURNAME_STR = "Surname"
```

String for default surname.

#### 13.9.2.6 UNKNOWN\_STR

```
const std::string person::UNKNOWN_STR = "Unknown"
```

String for unknown tags.

## 13.10 relation Namespace Reference

Namespace for strings for default relations.

## Variables

- `const std::string FATHERHOOD = "Fatherhood"`  
*String for Fatherhood.*
- `const std::string FATHERHOOD_DESC = "Relation between father and his child."`  
*String for Fatherhood description.*
- `const std::string FATHER = "Father"`  
*String for father.*
- `const std::string CHILD = "Child"`  
*String for child.*
- `const std::string MOTHERHOOD = "Motherhood"`  
*String for Motherhood.*
- `const std::string MOTHERHOOD_DESC = "Relation between mother and her child."`  
*String for Motherhood description.*
- `const std::string MOTHER = "Mother"`  
*String for mother.*
- `const std::string PARTNERSHIP = "Partnership"`  
*String for Partnership.*
- `const std::string PARTNERSHIP_DESC = "Relation between two partners."`  
*String for description of partnership.*
- `const std::string PARTNER = "Partner"`  
*String for partner.*
- `const std::string MARRIAGE = "Marriage"`  
*String for Marriage.*
- `const std::string MARRIAGE_DESC = "Bound by law between two people. Usually a man (husband) and a woman (wife)."`  
*String for Marriage description.*
- `const std::string WIFE = "Wife"`  
*String for wife.*
- `const std::string HUSBAND = "Husband"`  
*String for husband.*
- `const std::string ADOPTIVE_FATHERHOOD = "Adoptive fatherhood"`  
*String for title for adoptive fatherhood.*
- `const std::string DESC_ADOPTIVE_FATHERHOOD = "Fatherhood provided by adopting a child."`  
*String for description for adoptive fatherhood.*
- `const std::string ADOPTIVE_FATHER = "Adoptive father"`  
*String for adoptive father.*
- `const std::string ADOPTED_CHILD = "Adopted child"`  
*String for adopted child.*
- `const std::string ADOPTIVE_MOTHERHOOD = "Adoptive motherhood"`  
*String for title for adoptive motherhood.*
- `const std::string DESC_ADOPTIVE_MOTHERHOOD = "Motherhood provided by adopting a child."`  
*String for description for adoptive motherhood.*
- `const std::string ADOPTIVE_MOTHER = "Adoptive mother"`  
*String for adoptive father.*
- `const std::string BROTHERHOOD = "Brotherhood"`  
*String for brotherhood.*
- `const std::string DESC_BROTHERHOOD = "Relation between two brothers."`  
*String for brotherhood description.*
- `const std::string BROTHER = "Brother"`  
*String for brother.*

- `const std::string SISTERHOOD = "Sisterhood"`  
*String for sisterhood.*
- `const std::string DESC_SISTERHOOD = "Relation between two sisters."`  
*String for sisterhood description.*
- `const std::string SISTER = "Sister"`  
*String for sister.*
- `const std::string SIBLINGS = "Siblings"`  
*String for siblings.*
- `const std::string DESC_SIBLINGS = "Relation between brother and sister."`  
*String for siblings description.*

### 13.10.1 Detailed Description

Namespace for strings for default relations.

### 13.10.2 Variable Documentation

#### 13.10.2.1 ADOPTED\_CHILD

```
const std::string relation::ADOPTED_CHILD = "Adopted child"
```

String for adopted child.

#### 13.10.2.2 ADOPTIVE\_FATHER

```
const std::string relation::ADOPTIVE_FATHER = "Adoptive father"
```

String for adoptive father.

#### 13.10.2.3 ADOPTIVE\_FATHERHOOD

```
const std::string relation::ADOPTIVE_FATHERHOOD = "Adoptive fatherhood"
```

String for title for adoptive fatherhood.

#### 13.10.2.4 ADOPTIVE\_MOTHER

```
const std::string relation::ADOPTIVE_MOTHER = "Adoptive mother"
```

String for adoptive father.

#### 13.10.2.5 ADOPTIVE\_MOTHERHOOD

```
const std::string relation::ADOPTIVE_MOTHERHOOD = "Adoptive motherhood"
```

String for title for adoptive motherhood.

### 13.10.2.6 BROTHER

```
const std::string relation::BROTHER = "Brother"
```

String for brother.

### 13.10.2.7 BROTHERHHOD

```
const std::string relation::BROTHERHHOD = "Brotherhood"
```

String for brotherhood.

### 13.10.2.8 CHILD

```
const std::string relation::CHILD = "Child"
```

String for child.

### 13.10.2.9 DESC\_ADOPTIVE\_FATHERHOOD

```
const std::string relation::DESC_ADOPTIVE_FATHERHOOD = "Fatherhood provided by adopting a child."
```

String for description for adoptive fatherhood.

### 13.10.2.10 DESC\_ADOPTIVE\_MOTHERHOOD

```
const std::string relation::DESC_ADOPTIVE_MOTHERHOOD = "Motherhood provided by adopting a child."
```

String for description for adoptive motherhood.

### 13.10.2.11 DESC\_BROTHERHOOD

```
const std::string relation::DESC_BROTHERHOOD = "Relation between two brothers."
```

String for brotherhood description.

### 13.10.2.12 DESC\_SIBLINGS

```
const std::string relation::DESC_SIBLINGS = "Relation between brother and sister."
```

String for siblings description.

#### 13.10.2.13 DESC\_SISTERHOOD

```
const std::string relation::DESC_SISTERHOOD = "Relation between two sisters."
```

String for sisterhood description.

#### 13.10.2.14 FATHER

```
const std::string relation::FATHER = "Father"
```

String for father.

#### 13.10.2.15 FATHERHOOD

```
const std::string relation::FATHERHOOD = "Fatherhood"
```

String for Fatherhood.

#### 13.10.2.16 FATHERHOOD\_DESC

```
const std::string relation::FATHERHOOD_DESC = "Relation between father and his child."
```

String for Fatherhood description.

#### 13.10.2.17 HUSBAND

```
const std::string relation::HUSBAND = "Husband"
```

String for husband.

#### 13.10.2.18 MARRIAGE

```
const std::string relation::MARRIAGE = "Marriage"
```

String for Marriage.

#### 13.10.2.19 MARRIAGE\_DESC

```
const std::string relation::MARRIAGE_DESC = "Bound by law between two people. Usually a man  
(husband) and a woman (wife)."
```

String for Marriage description.

**13.10.2.20 MOTHER**

```
const std::string relation::MOTHER = "Mother"
```

String for mother.

**13.10.2.21 MOTHERHOOD**

```
const std::string relation::MOTHERHOOD = "Motherhood"
```

String for Motherhood.

**13.10.2.22 MOTHERHOOD\_DESC**

```
const std::string relation::MOTHERHOOD_DESC = "Relation between mother and her child."
```

String for Motherhood description.

**13.10.2.23 PARTNER**

```
const std::string relation::PARTNER = "Partner"
```

String for partner.

**13.10.2.24 PARTNERSHIP**

```
const std::string relation::PARTNERSHIP = "Partnership"
```

String for Partnership.

**13.10.2.25 PARTNERSHIP\_DESC**

```
const std::string relation::PARTNERSHIP_DESC = "Relation between two partners."
```

String for description of partnership.

**13.10.2.26 SIBLINGS**

```
const std::string relation::SIBLINGS = "Siblings"
```

String for siblings.

**13.10.2.27 SISTER**

```
const std::string relation::SISTER = "Sister"
```

String for sister.

### 13.10.2.28 SISTERHOOD

```
const std::string relation::SISTERHOOD = "Sisterhood"
```

String for sisterhood.

### 13.10.2.29 WIFE

```
const std::string relation::WIFE = "Wife"
```

String for wife.

## 13.11 tagchooser Namespace Reference

Namespace for strings shown in tag chooser dialog.

### Variables

- const std::string [CUSTOM\\_TAG](#) = "+ Custom tag"  
*Last custom tag choice.*

### 13.11.1 Detailed Description

Namespace for strings shown in tag chooser dialog.

### 13.11.2 Variable Documentation

#### 13.11.2.1 CUSTOM\_TAG

```
const std::string tagchooser::CUSTOM_TAG = "+ Custom tag"
```

Last custom tag choice.

## 13.12 trait Namespace Reference

Namespace for everything with relation traits.

### Functions

- std::string [str](#) ([Trait](#) t)

## Variables

- constexpr [Trait](#) AllTraits [] = {[None](#), [Fatherhood](#), [Motherhood](#), [Partnership](#), [Sibling](#)}  
*All possible traits.*

### 13.12.1 Detailed Description

Namespace for everything with relation traits.

### 13.12.2 Function Documentation

#### 13.12.2.1 str()

```
std::string trait::str (  
    Trait t )
```

Represent trait as a string.

#### Parameters

|          |              |
|----------|--------------|
| <i>t</i> | Given trait. |
|----------|--------------|

#### Returns

String representing given trait.

### 13.12.3 Variable Documentation

#### 13.12.3.1 AllTraits

```
constexpr Trait trait::AllTraits[] = {None, Fatherhood, Motherhood, Partnership, Sibling}  
[constexpr]
```

All possible traits.

## 13.13 Ui Namespace Reference

Namespace for classes having all its user interface items.

### 13.13.1 Detailed Description

Namespace for classes having all its user interface items.



# Chapter 14

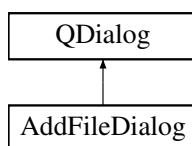
## Class Documentation

### 14.1 AddFileDialog Class Reference

Dialog for adding file.

```
#include <dialogs.h>
```

Inheritance diagram for AddFileDialog:



#### Public Slots

- void [copyFile](#) ()  
*Open file chooser for copying file.*
- void [createOrChooseFile](#) ()  
*Create or choose file based on selected tab.*
- void [openFile](#) (QTreeWidgetItem \*item)

#### Public Member Functions

- [AddFileDialog](#) ([FileType](#) type, const std::vector< std::string > &extensions, [VirtualDrive](#) \*parentDrive, [FamilyTree](#) \*FT, QWidget \*parent)
- [~AddFileDialog](#) ()  
*Default destructor.*

#### Private Member Functions

- size\_t [getId](#) (QTreeWidgetItem \*item)

## Private Attributes

- [FamilyTree](#) \* `FT_`  
*Used family tree.*
- [VirtualDrive](#) \* `drive_`  
*Parent drive.*
- [FileType](#) `type_`  
*Type of the file.*
- [Ui::AddFileDialog](#) \* `ui`  
*User interface of the Qt framework.*

### 14.1.1 Detailed Description

Dialog for adding file.

### 14.1.2 Constructor & Destructor Documentation

#### 14.1.2.1 AddFileDialog()

```
AddFileDialog::AddFileDialog (
    FileType type,
    const std::vector< std::string > & extensions,
    VirtualDrive * parentDrive,
    FamilyTree * FT,
    QWidget * parent )
```

Default constructor.

#### Parameters

|                    |   |
|--------------------|---|
| <i>type</i>        | If the file is general file, media or note. |
| <i>extensions</i>  | Vector of all extensions to show.           |
| <i>parentDrive</i> | Parent drive where the file will be stored. |
| <i>FT</i>          | Given family tree the current app is using. |
| <i>parent</i>      | The Qt Widget parent.                       |

#### 14.1.2.2 ~AddFileDialog()

```
AddFileDialog::~AddFileDialog ( )
```

Default destructor.

### 14.1.3 Member Function Documentation

#### 14.1.3.1 copyFile

```
void AddFileDialog::copyFile ( ) [slot]
```

Open file chooser for copying file.

### 14.1.3.2 createOrChooseFile

```
void AddFileDialog::createOrChooseFile ( ) [slot]
```

Create or choose file based on selected tab.

### 14.1.3.3 getId()

```
size_t AddFileDialog::getId (
    QTreeWidgetItem * item ) [private]
```

Get the id from the text of the given id.

#### Parameters

|             |                        |
|-------------|------------------------|
| <i>item</i> | Which item has the id. |
|-------------|------------------------|

#### Returns

Id from its text.

### 14.1.3.4 openFile

```
void AddFileDialog::openFile (
    QTreeWidgetItem * item ) [slot]
```

Show file upon clicking it.

#### Parameters

|             |                         |
|-------------|-------------------------|
| <i>item</i> | Which item was clicked. |
|-------------|-------------------------|

## 14.1.4 Member Data Documentation

### 14.1.4.1 drive\_

```
VirtualDrive* AddFileDialog::drive_ [private]
```

Parent drive.

### 14.1.4.2 FT\_

```
FamilyTree* AddFileDialog::FT_ [private]
```

Used family tree.

#### 14.1.4.3 type\_

```
FileType AddFileDialog::type_ [private]
```

Type of the file.

#### 14.1.4.4 ui

```
Ui::AddFileDialog* AddFileDialog::ui [private]
```

User interface of the Qt framework.

The documentation for this class was generated from the following files:

- [graphics/dialogs.h](#)
- [graphics/dialogs.cpp](#)

## 14.2 AppSettings Struct Reference

Struct for app settings.

```
#include <config.h>
```

### Public Member Functions

- void [readJson](#) (const Json::Value &value)
- void [setDefault](#) ()  
*Set the default options.*

### Public Attributes

- std::string [fontFamily](#)  
*Used font in the graphic boxes.*
- int [fontSize](#)  
*Size of the font used in the graphic boxes.*
- int [genSizeDown](#)  
*Number of generations to show down.*
- int [genSizeUp](#)  
*Number of generations to show up.*
- Color [highlightedColor](#)  
*Color for highlighted box.*
- Color [lineColor](#)  
*Color of the lines.*
- int [lineWidth](#)  
*Width of the line.*
- Color [probandColor](#)  
*Color for proband.*
- Color [promoted](#)

- [Color](#) for promoted relation.
- int [radius](#)  
*Border radius.*
- bool [showEventTab](#)  
*If event tab should be visible.*
- bool [showFileTab](#)  
*If file tab should be visible.*
- bool [showGeneralTab](#)  
*If general tab should be visible.*
- bool [showMediaTab](#)  
*If media tab should be visible.*
- bool [showNoteTab](#)  
*If note tab should be visible.*
- bool [showRelationTab](#)  
*If relation tab should be visible.*
- int [sizeX](#)  
*Size of the box in x axis.*
- int [sizeY](#)  
*Size of the box in y axis.*
- int [splitterPositionOne](#)  
*Splitter position one.*
- int [splitterPositionTwo](#)  
*Splitter position two.*
- [Color](#) [standard](#)  
*Standard color.*
- [TabPosition](#) [tabPosition](#)  
*Tab position of the information tab.*
- [Color](#) [textColor](#)  
*Color of the text.*

### 14.2.1 Detailed Description

Struct for app settings.

### 14.2.2 Member Function Documentation

#### 14.2.2.1 readJson()

```
void AppSettings::readJson (
    const Json::Value & value )
```

Read and load data from JSON value.

Parameters

|              |                   |
|--------------|-------------------|
| <i>value</i> | Given JSON value. |
|--------------|-------------------|

#### 14.2.2.2 `setDefault()`

```
void AppSettings::setDefault ( )
```

Set the default options.

### 14.2.3 Member Data Documentation

#### 14.2.3.1 `fontFamily`

```
std::string AppSettings::fontFamily
```

Used font in the graphic boxes.

#### 14.2.3.2 `fontSize`

```
int AppSettings::fontSize
```

Size of the font used in the graphic boxes.

#### 14.2.3.3 `genSizeDown`

```
int AppSettings::genSizeDown
```

Number of generations to show down.

#### 14.2.3.4 `genSizeUp`

```
int AppSettings::genSizeUp
```

Number of generations to show up.

#### 14.2.3.5 `highlightedColor`

```
Color AppSettings::highlightedColor
```

[Color](#) for highlighted box.

#### 14.2.3.6 `lineColor`

```
Color AppSettings::lineColor
```

[Color](#) of the lines.

### 14.2.3.7 lineWidth

```
int AppSettings::lineWidth
```

Width of the line.

### 14.2.3.8 probandColor

```
Color AppSettings::probandColor
```

Color for proband.

### 14.2.3.9 promoted

```
Color AppSettings::promoted
```

Color for promoted relation.

### 14.2.3.10 radius

```
int AppSettings::radius
```

Border radius.

### 14.2.3.11 showEventTab

```
bool AppSettings::showEventTab
```

If event tab should be visible.

### 14.2.3.12 showFileTab

```
bool AppSettings::showFileTab
```

If file tab should be visible.

### 14.2.3.13 showGeneralTab

```
bool AppSettings::showGeneralTab
```

If general tab should be visible.

### 14.2.3.14 showMediaTab

```
bool AppSettings::showMediaTab
```

If media tab should be visible.

#### 14.2.3.15 showNoteTab

```
bool AppSettings::showNoteTab
```

If note tab should be visible.

#### 14.2.3.16 showRelationTab

```
bool AppSettings::showRelationTab
```

If relation tab should be visible.

#### 14.2.3.17 sizeX

```
int AppSettings::sizeX
```

Size of the box in x axis.

#### 14.2.3.18 sizeY

```
int AppSettings::sizeY
```

Size of the box in y axis.

#### 14.2.3.19 splitterPositionOne

```
int AppSettings::splitterPositionOne
```

Splitter position one.

#### 14.2.3.20 splitterPositionTwo

```
int AppSettings::splitterPositionTwo
```

Splitter position two.

#### 14.2.3.21 standard

```
Color AppSettings::standard
```

Standard color.

#### 14.2.3.22 tabPosition

```
TabPosition AppSettings::tabPosition
```

Tab position of the information tab.



### 14.2.3.23 textColor

`Color` `AppSettings::textColor`

`Color` of the text.

The documentation for this struct was generated from the following files:

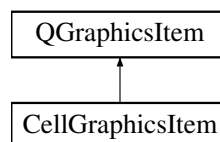
- `core/config.h`
- `core/config.cpp`

## 14.3 CellGraphicsItem Class Reference

Cell around all relations.

```
#include <graphics_items.h>
```

Inheritance diagram for CellGraphicsItem:



### Public Member Functions

- `CellGraphicsItem` (int `sizeX`, int `sizeY`, qreal `x`, qreal `y`, `QPen` \*`pen`, int `borderRadius`)
- `QRectF` `boundingRect` () const override
- void `paint` (`QPainter` \*`painter`, const `QStyleOptionGraphicsItem` \*`option`, `QWidget` \*`widget`) override
- `QPainterPath` `shape` () const override

### Private Attributes

- int `borderRadius_`  
*Border radius.*
- `QPen` \* `pen_`  
*Pointer to the used pen.*
- int `sizeX_`  
*Size of the cell in the x axis.*
- int `sizeY_`  
*Size of the cell in the y axis.*
- qreal `x_`  
*Top left point x position.*
- qreal `y_`  
*Top left point y position.*

### 14.3.1 Detailed Description

Cell around all relations.

### 14.3.2 Constructor & Destructor Documentation

#### 14.3.2.1 CellGraphicsItem()

```
CellGraphicsItem::CellGraphicsItem (
    int sizeX,
    int sizeY,
    qreal x,
    qreal y,
    QPen * pen,
    int borderRadius )
```

Default constructor.

##### Parameters

|                     |                                   |
|---------------------|-----------------------------------|
| <i>sizeX</i>        | Size of the cell in the x axis.   |
| <i>sizeY</i>        | Size of the cell in the y axis.   |
| <i>x</i>            | Top left point x position.        |
| <i>y</i>            | Top left point y position.        |
| <i>pen</i>          | Which pen is used for lines.      |
| <i>borderRadius</i> | What is the radius of the border. |

### 14.3.3 Member Function Documentation

#### 14.3.3.1 boundingRect()

```
QRectF CellGraphicsItem::boundingRect ( ) const [override]
```

Get the bounding rectangle of the object.

##### Returns

Rectangle that is bounding the item.

#### 14.3.3.2 paint()

```
void CellGraphicsItem::paint (
    QPainter * painter,
    const QStyleOptionGraphicsItem * option,
    QWidget * widget ) [override]
```

Paint the object to its painter.

**Parameters**

|                |   |
|----------------|---|
| <i>painter</i> | Which painter to use to paint the object. |
| <i>option</i>  | Option for the Qt framework (not used).   |
| <i>widget</i>  | Widget for the Qt framework (not used).   |

**14.3.3.3 shape()**

```
QPainterPath CellGraphicsItem::shape ( ) const [override]
```

Get the shape of the item.

**Returns**

Get the shape to paint.

**14.3.4 Member Data Documentation****14.3.4.1 borderRadius\_**

```
int CellGraphicsItem::borderRadius_ [private]
```

Border radius.

**14.3.4.2 pen\_**

```
QPen* CellGraphicsItem::pen_ [private]
```

Pointer to the used pen.

**14.3.4.3 sizex\_**

```
int CellGraphicsItem::sizex_ [private]
```

Size of the cell in the x axis.

**14.3.4.4 sizey\_**

```
int CellGraphicsItem::sizey_ [private]
```

Size of the cell in the y axis.

**14.3.4.5 x\_**

```
qreal CellGraphicsItem::x_ [private]
```

Top left point x position.

#### 14.3.4.6 y\_

```
qreal CellGraphicsItem::y_ [private]
```

Top left point y position.

The documentation for this class was generated from the following files:

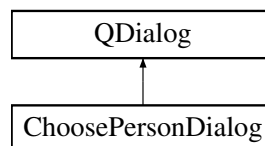
- [graphics/graphics\\_items.h](#)
- [graphics/graphics\\_items.cpp](#)

## 14.4 ChoosePersonDialog Class Reference

Dialog for choosing person with find window. Parent have to have slot [savePerson\(\)](#).

```
#include <dialogs.h>
```

Inheritance diagram for ChoosePersonDialog:



### Public Slots

- void [clearFilter\(\)](#)  
*Clear filter and show all persons.*
- void [filterPersons\(\)](#)  
*Filter persons by the text edit.*
- void [saveSelectedPerson\(\)](#)  
*Save currently selected person.*

### Signals

- void [savePerson\(Person \\*person\)](#)  
*When save can be proceeded.*

### Public Member Functions

- [ChoosePersonDialog\(FamilyTree \\*FT, Person \\*person, QWidget \\*parent=nullptr\)](#)
- [~ChoosePersonDialog\(\)](#)  
*Default destructor.*

### Private Attributes

- [FamilyTree](#) \* [FT\\_](#)  
*Pointer to the family tree.*
- [Person](#) \* [person\\_](#)  
*Pointer for.*
- [Ui::ChoosePersonDialog](#) \* [ui](#)  
*User interface of the Qt framework.*

## 14.4.1 Detailed Description

Dialog for choosing person with find window. Parent have to have slot [savePerson\(\)](#).

## 14.4.2 Constructor & Destructor Documentation

### 14.4.2.1 ChoosePersonDialog()

```
ChoosePersonDialog::ChoosePersonDialog (
    FamilyTree * FT,
    Person * person,
    QWidget * parent = nullptr )
```

Default constructor.

#### Parameters

|               |  |
|---------------|--|
| <i>FT</i>     | Pointer to the family tree.  |
| <i>person</i> | Pointer to the person to be replaced or nullptr to choose new one. |
| <i>parent</i> | The Qt Widget parent.  |

### 14.4.2.2 ~ChoosePersonDialog()

```
ChoosePersonDialog::~~ChoosePersonDialog ( )
```

Default destructor.

## 14.4.3 Member Function Documentation

### 14.4.3.1 clearFilter

```
void ChoosePersonDialog::clearFilter ( ) [slot]
```

Clear filter and show all persons.

#### 14.4.3.2 filterPersons

```
void ChoosePersonDialog::filterPersons ( ) [slot]
```

Filter persons by the text edit.

#### 14.4.3.3 savePerson

```
void ChoosePersonDialog::savePerson (
    Person * person ) [signal]
```

When save can be proceeded.

#### 14.4.3.4 saveSelectedPerson

```
void ChoosePersonDialog::saveSelectedPerson ( ) [slot]
```

Save currently selected person.

### 14.4.4 Member Data Documentation

#### 14.4.4.1 FT\_

```
FamilyTree* ChoosePersonDialog::FT_ [private]
```

Pointer to the family tree.

#### 14.4.4.2 person\_

```
Person* ChoosePersonDialog::person_ [private]
```

Pointer for.

#### 14.4.4.3 ui

```
Ui::ChoosePersonDialog* ChoosePersonDialog::ui [private]
```

User interface of the Qt framework.

The documentation for this class was generated from the following files:

- [graphics/dialogs.h](#)
- [graphics/dialogs.cpp](#)

## 14.5 Color Struct Reference

Struct for general color.

```
#include <config.h>
```

### Public Member Functions

- [Color](#) (int r=255, int g=255, int b=255)
- void [readJson](#) (const Json::Value &value)  
*Read it from json value.*

### Public Attributes

- int [red](#)  
*Red part of the color.*
- int [green](#)  
*Green part of the color.*
- int [blue](#)  
*Blue part of the color.*

### 14.5.1 Detailed Description

Struct for general color.

### 14.5.2 Constructor & Destructor Documentation

#### 14.5.2.1 Color()

```
Color::Color (  
    int r = 255,  
    int g = 255,  
    int b = 255 ) [explicit]
```

Default constructor.

#### Parameters

|          |        |
|----------|--------|
| <i>r</i> | Red.   |
| <i>g</i> | Green. |
| <i>b</i> | Blue.  |

### 14.5.3 Member Function Documentation

#### 14.5.3.1 readJson()

```
void Color::readJson (
    const Json::Value & value )
```

Read it from json value.

### 14.5.4 Member Data Documentation

#### 14.5.4.1 blue

```
int Color::blue
```

Blue part of the color.

#### 14.5.4.2 green

```
int Color::green
```

Green part of the color.

#### 14.5.4.3 red

```
int Color::red
```

Red part of the color.

The documentation for this struct was generated from the following files:

- [core/config.h](#)
- [core/config.cpp](#)

## 14.6 Date Class Reference

Basic date format. Holding three numbers for year, month and day. If some of the numbers are not defined 0 is present.

```
#include <date.h>
```



## Public Member Functions

- `Date` (int y=0, int m=0, int d=0)
- int `getDay` () const
- int `getMonth` () const
- int `getYear` () const
- bool `isEmpty` () const
- bool `isExact` () const
- void `read` (const std::string &date)
- void `setDay` (int d)
- void `setMonth` (int m)
- void `setYear` (int y)
- std::string `str` () const

## Private Attributes

- int `day`  
*Day number.*
- int `month`  
*Month number.*
- int `year`  
*Year number.*

### 14.6.1 Detailed Description

Basic date format. Holding three numbers for year, month and day. If some of the numbers are not defined 0 is present.

### 14.6.2 Constructor & Destructor Documentation

#### 14.6.2.1 Date()

```
Date::Date (  
    int y = 0,  
    int m = 0,  
    int d = 0 ) [explicit]
```

Default constructor.

#### Parameters

|          |               |
|----------|---------------|
| <i>d</i> | Day number.   |
| <i>m</i> | Month number. |
| <i>y</i> | Year number.  |

### 14.6.3 Member Function Documentation

#### 14.6.3.1 `getDay()`

```
int Date::getDay ( ) const
```

Get the day number.

##### Returns

Day number (0 if it was not set).

#### 14.6.3.2 `getMonth()`

```
int Date::getMonth ( ) const
```

Get the month number.

##### Returns

Month number (0 if it was not set).

#### 14.6.3.3 `getYear()`

```
int Date::getYear ( ) const
```

Get the year number.

##### Returns

Year number (0 if it was not set).

#### 14.6.3.4 `isEmpty()`

```
bool Date::isEmpty ( ) const
```

Check if the date is completely not defined. All values are non-defined if they are equal to 0.

##### Returns

True if all values are 0.

#### 14.6.3.5 `isExact()`

```
bool Date::isExact ( ) const
```

Check if the date is exact or not. [Date](#) is exact when all values are defined (non-zero).

##### Returns

True if the date is exact.

#### 14.6.3.6 `read()`

```
void Date::read (
    const std::string & date )
```

Load date from string in specified flexible format: yyyy-mm-dd (if there would be for example 05 it can only be 5). If it does not follow the format there will be corrupted data and won't raise an exception.

## Parameters

|             |                                 |
|-------------|---------------------------------|
| <i>date</i> | Written date in correct format. |
|-------------|---------------------------------|

**14.6.3.7 setDay()**

```
void Date::setDay (
    int d )
```

Set the day number.

## Parameters

|          |             |
|----------|-------------|
| <i>d</i> | Day number. |
|----------|-------------|

**14.6.3.8 setMonth()**

```
void Date::setMonth (
    int m )
```

Set the month number.

## Parameters

|          |               |
|----------|---------------|
| <i>m</i> | Month number. |
|----------|---------------|

**14.6.3.9 setYear()**

```
void Date::setYear (
    int y )
```

Set the year number.

## Parameters

|          |              |
|----------|--------------|
| <i>y</i> | Year number. |
|----------|--------------|

**14.6.3.10 str()**

```
std::string Date::str ( ) const
```

Get string representing the date in format yyyy-MM-DD. If one of the values are set to 0, then it is replaced by the same number of ?

## Returns

String in the format.

## 14.6.4 Member Data Documentation

### 14.6.4.1 day

```
int Date::day [private]
```

Day number.

### 14.6.4.2 month

```
int Date::month [private]
```

Month number.

### 14.6.4.3 year

```
int Date::year [private]
```

Year number.

The documentation for this class was generated from the following files:

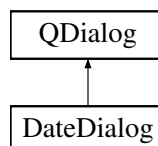
- [core/date.h](#)
- [core/date.cpp](#)

## 14.7 QDialog Class Reference

Dialog for editing custom dates.

```
#include <dialogs.h>
```

Inheritance diagram for QDialog:



### Public Slots

- void [saveDate](#) ()  
*Slot for saving the date if OK button was pressed.*
- void [unknownDate](#) (bool isUnknown)

## Public Member Functions

- [DateDialog](#) ([FamilyTree](#) \*FT, [WrappedDate](#) \*date=nullptr, [QWidget](#) \*parent=nullptr)
- [~DateDialog](#) ()

*Default destructor.*

## Private Attributes

- [WrappedDate](#) \* [date\\_](#)  
*Pointer to the used date.*
- [FamilyTree](#) \* [FT\\_](#)  
*Pointer to the family tree.*
- [Ui::DateDialog](#) \* [ui](#)  
*User interface of the Qt framework.*

## 14.7.1 Detailed Description

Dialog for editing custom dates.

## 14.7.2 Constructor & Destructor Documentation

### 14.7.2.1 DateDialog()

```
DateDialog::DateDialog (
    FamilyTree * FT,
    WrappedDate * date = nullptr,
    QWidget * parent = nullptr ) [explicit]
```

Default constructor.

#### Parameters

|               |                             |
|---------------|-----------------------------|
| <i>FT</i>     | Pointer to the family tree. |
| <i>date</i>   | Pointer to the used date.   |
| <i>parent</i> | The Qt Widget parent.       |

### 14.7.2.2 ~DateDialog()

```
DateDialog::~DateDialog ( )
```

Default destructor.

## 14.7.3 Member Function Documentation

### 14.7.3.1 saveDate

```
void DateDialog::saveDate ( ) [slot]
```

Slot for saving the date if OK button was pressed.

### 14.7.3.2 unknownDate

```
void DateDialog::unknownDate (
    bool isUnknown ) [slot]
```

If the unknown check box was changed.

#### Parameters

|                  |                               |
|------------------|-------------------------------|
| <i>isUnknown</i> | If the check box was checked. |
|------------------|-------------------------------|

## 14.7.4 Member Data Documentation

### 14.7.4.1 date\_

```
WrappedDate* DateDialog::date_ [private]
```

Pointer to the used date.

### 14.7.4.2 FT\_

```
FamilyTree* DateDialog::FT_ [private]
```

Pointer to the family tree.

### 14.7.4.3 ui

```
Ui::DateDialog* DateDialog::ui [private]
```

User interface of the Qt framework.

The documentation for this class was generated from the following files:

- [graphics/dialogs.h](#)
- [graphics/dialogs.cpp](#)

## 14.8 Event Class Reference

Class holding data for a single event.

```
#include <family_tree_items.h>
```

## Public Member Functions

- [Event](#) ([Settings](#) \*settings)
- void [addPerson](#) (size\_t person, const std::string &role=[events::NO\\_ROLE](#))
- void [clearPersons](#) ()
  - Clear all persons.*
- [WrappedDate](#) \* [date](#) ()
- const [WrappedDate](#) & [getDate](#) () const
- size\_t [getId](#) () const
- const std::vector< std::pair< std::string, size\_t > > & [getPersons](#) () const
- const std::string & [getPlace](#) () const
- const [Settings](#) & [getSettings](#) () const
- size\_t [getTemplate](#) () const
- const std::string & [getText](#) () const
- void [printHtml](#) (std::ostream &os, const std::vector< std::pair< std::string, std::string > > &persons)
- void [readJson](#) (const Json::Value &value)
- bool [removePerson](#) (size\_t id)
- [WrappedDate](#) & [setDate](#) ()
- void [setId](#) (size\_t id)
- void [setPlace](#) (const std::string &place)
- void [setTemplate](#) (size\_t templ)
- void [setText](#) (const std::string &text)
- std::string [str](#) () const
- std::vector< size\_t > [updateToTemplate](#) ()

## Private Attributes

- [WrappedDate](#) [date\\_](#)
  - Date of this event.*
- size\_t [id\\_](#)
  - Id of this event.*
- std::vector< std::pair< std::string, size\_t > > [persons\\_](#)
  - People with their roles.*
- std::string [place\\_](#)
  - Place of this event.*
- [Settings](#) \* [settings\\_](#)
  - Settings used in app.*
- size\_t [template\\_](#)
  - Template for this event.*
- std::string [text\\_](#)
  - Text (note or value) for this event.*

### 14.8.1 Detailed Description

Class holding data for a single event.

### 14.8.2 Constructor & Destructor Documentation

#### 14.8.2.1 Event()

```
Event::Event (
    Settings * settings ) [explicit]
```

Default constructor.

## Parameters

|                 |  |
|-----------------|--|
| <i>settings</i> | The settings used in app (to access the template). |
|-----------------|--|

## 14.8.3 Member Function Documentation

### 14.8.3.1 addPerson()

```
void Event::addPerson (
    size_t person,
    const std::string & role = events::NO_ROLE )
```

Add person to the vector.

## Parameters

|               |  |
|---------------|--|
| <i>person</i> | Id of the person.  |
| <i>role</i>   | Special role for this person, or empty if he does not have special role. |

### 14.8.3.2 clearPersons()

```
void Event::clearPersons ( )
```

Clear all persons.

### 14.8.3.3 date()

```
WrappedDate * Event::date ( )
```

Get the pointer to the date.

## Returns

Pointer to the wrapped date.

### 14.8.3.4 getDate()

```
const WrappedDate & Event::getDate ( ) const
```

Get the date of this event.

## Returns

Constant reference to the date.



#### 14.8.3.5 getId()

```
size_t Event::getId ( ) const
```

Get the id of this event.

##### Returns

Id of the event.

#### 14.8.3.6 getPersons()

```
const std::vector< std::pair< std::string, size_t > > & Event::getPersons ( ) const
```

Get the vector of persons.

##### Returns

Constant reference to the vector of persons.

#### 14.8.3.7 getPlace()

```
const std::string & Event::getPlace ( ) const
```

Get the place of this event.

##### Returns

Constant reference to the place.

#### 14.8.3.8 getSettings()

```
const Settings & Event::getSettings ( ) const
```

Get the settings.

##### Returns

Constant reference to the settings.

#### 14.8.3.9 getTemplate()

```
size_t Event::getTemplate ( ) const
```

Get the template for this event.

##### Returns

Id of the template.

#### 14.8.3.10 getText()

```
const std::string & Event::getText ( ) const
```

Get the text note of the event.

##### Returns

Constant reference to the text.

#### 14.8.3.11 printHtml()

```
void Event::printHtml (
    std::ostream & os,
    const std::vector< std::pair< std::string, std::string > > & persons )
```

Print event to an output stream in HTML format.

##### Parameters

|                |                                      |
|----------------|--------------------------------------|
| <i>os</i>      | Given output stream,                 |
| <i>persons</i> | List of pairs for role&person names. |

#### 14.8.3.12 readJson()

```
void Event::readJson (
    const Json::Value & value )
```

Read the relation from JSON value.

##### Parameters

|              |                           |
|--------------|---------------------------|
| <i>value</i> | The JSON value from file. |
|--------------|---------------------------|

#### 14.8.3.13 removePerson()

```
bool Event::removePerson (
    size_t id )
```

Remove person from the list of all persons.

##### Parameters

|           |                   |
|-----------|-------------------|
| <i>id</i> | Id of the person. |
|-----------|-------------------|

**Returns**

True if there is no other person left (thus left out event), otherwise false.

**14.8.3.14 setDate()**

```
WrappedDate & Event::setDate ( )
```

Set the date for this event.

**Returns**

Reference to the date.

**14.8.3.15 setId()**

```
void Event::setId (
    size_t id )
```

Set the id for this event.

**Parameters**

|           |           |
|-----------|-----------|
| <i>id</i> | Given id. |
|-----------|-----------|

**14.8.3.16 setPlace()**

```
void Event::setPlace (
    const std::string & place )
```

Set the place for this event.

**Parameters**

|              |            |
|--------------|------------|
| <i>place</i> | New place. |
|--------------|------------|

**14.8.3.17 setTemplate()**

```
void Event::setTemplate (
    size_t templ )
```

Set the template of this event.

**Parameters**

|              |                         |
|--------------|-------------------------|
| <i>templ</i> | Id of the new template. |
|--------------|-------------------------|

#### 14.8.3.18 setText()

```
void Event::setText (
    const std::string & text )
```

Set the text for this event.

##### Parameters

|             |                 |
|-------------|-----------------|
| <i>text</i> | Given new text. |
|-------------|-----------------|

#### 14.8.3.19 str()

```
std::string Event::str ( ) const
```

Get the event representation as a string.

##### Returns

Newly constructed string representing the event.

#### 14.8.3.20 updateToTemplate()

```
std::vector< size_t > Event::updateToTemplate ( )
```

Update this event to the changed template.

##### Returns

Vector of all persons which were removed.

### 14.8.4 Member Data Documentation

#### 14.8.4.1 date\_

```
WrappedDate Event::date_ [private]
```

[Date](#) of this event.

#### 14.8.4.2 id\_

```
size_t Event::id_ [private]
```

Id of this event.

#### 14.8.4.3 persons\_

```
std::vector<std::pair<std::string, size_t> > Event::persons_ [private]
```

People with their roles.

#### 14.8.4.4 place\_

```
std::string Event::place_ [private]
```

Place of this event.

#### 14.8.4.5 settings\_

```
Settings* Event::settings_ [private]
```

[Settings](#) used in app.

#### 14.8.4.6 template\_

```
size_t Event::template_ [private]
```

Template for this event.

#### 14.8.4.7 text\_

```
std::string Event::text_ [private]
```

Text (note or value) for this event.

The documentation for this class was generated from the following files:

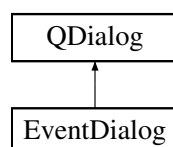
- [core/family\\_tree\\_items.h](#)
- [core/family\\_tree\\_items.cpp](#)

## 14.9 EventDialog Class Reference

Dialog showing single event for editing it.

```
#include <dialogs.h>
```

Inheritance diagram for EventDialog:



## Public Slots

- void [addEmptyRole](#) ()  
*Add empty role to the roles.*
- void [choosePerson](#) ()  
*Open dialog for choosing person.*
- void [openCustomDate](#) ()  
*Open dialog for custom date.*
- void [refreshDates](#) ()  
*Refresh date from dialog.*
- void [refreshPersons](#) (QTreeWidgetItem \*item)
- void [removeEmptyRole](#) ()  
*Remove empty role from the list.*
- void [removePerson](#) ()  
*Remove person from its role.*
- void [saveEvent](#) ()  
*Save the data to the event.*
- void [savePerson](#) (Person \*person)
- void [showTemplate](#) ()  
*Show the template details.*

## Public Member Functions

- [EventDialog](#) (FamilyTree \*FT, QWidget \*parent=nullptr, [Event](#) \*event=nullptr)
- [~EventDialog](#) ()  
*Default destructor.*

## Private Member Functions

- void [showEventDetails](#) ()  
*Show event details.*

## Private Attributes

- [WrappedDate](#) [date\\_](#)  
*Wrapped date of this event.*
- [Event](#) \* [event\\_](#)  
*Pointer to the event that is being edited, may be a null pointer.*
- [FamilyTree](#) \* [FT\\_](#)  
*Pointer to the family tree.*
- [Person](#) \* [main\\_](#)  
*Pointer to the main person used if no other persons are in event.*
- Ui::EventDialog \* [ui](#)  
*User interface of the Qt framework.*

### 14.9.1 Detailed Description

Dialog showing single event for editing it.

## 14.9.2 Constructor & Destructor Documentation

### 14.9.2.1 EventDialog()

```
EventDialog::EventDialog (
    FamilyTree * FT,
    QWidget * parent = nullptr,
    Event * event = nullptr ) [explicit]
```

Default constructor.

#### Parameters

|               |                                     |
|---------------|-------------------------------------|
| <i>FT</i>     | Pointer to the family tree.         |
| <i>parent</i> | The Qt Widget parent.               |
| <i>event</i>  | Which event is being edited if any. |

### 14.9.2.2 ~EventDialog()

```
EventDialog::~EventDialog ( )
```

Default destructor.

## 14.9.3 Member Function Documentation

### 14.9.3.1 addEmptyRole

```
void EventDialog::addEmptyRole ( ) [slot]
```

Add empty role to the roles.

### 14.9.3.2 choosePerson

```
void EventDialog::choosePerson ( ) [slot]
```

Open dialog for choosing person.

### 14.9.3.3 openCustomDate

```
void EventDialog::openCustomDate ( ) [slot]
```

Open dialog for custom date.

### 14.9.3.4 refreshDates

```
void EventDialog::refreshDates ( ) [slot]
```

Refresh date from dialog.

### 14.9.3.5 refreshPersons

```
void EventDialog::refreshPersons (
    QTreeWidgetItem * item ) [slot]
```

Refresh persons comboBox to the current role and its value.

**Parameters**

|             |                            |
|-------------|----------------------------|
| <i>item</i> | Which item was clicked on. |
|-------------|----------------------------|

**14.9.3.6 removeEmptyRole**

```
void EventDialog::removeEmptyRole ( ) [slot]
```

Remove empty role from the list.

**14.9.3.7 removePerson**

```
void EventDialog::removePerson ( ) [slot]
```

Remove person from its role.

**14.9.3.8 saveEvent**

```
void EventDialog::saveEvent ( ) [slot]
```

Save the data to the event.

**14.9.3.9 savePerson**

```
void EventDialog::savePerson (
    Person * person ) [slot]
```

Save person from dialog.

**Parameters**

|               |                            |
|---------------|----------------------------|
| <i>person</i> | Which person was selected. |
|---------------|----------------------------|

**14.9.3.10 showEventDetails()**

```
void EventDialog::showEventDetails ( ) [private]
```

Show event details.

**14.9.3.11 showTemplate**

```
void EventDialog::showTemplate ( ) [slot]
```

Show the template details.



## 14.9.4 Member Data Documentation

### 14.9.4.1 date\_

`WrappedDate` `EventDialog::date_` [private]

Wrapped date of this event.

### 14.9.4.2 event\_

`Event*` `EventDialog::event_` [private]

Pointer to the event that is being edited, may be a null pointer.

### 14.9.4.3 FT\_

`FamilyTree*` `EventDialog::FT_` [private]

Pointer to the family tree.

### 14.9.4.4 main\_

`Person*` `EventDialog::main_` [private]

Pointer to the main person used if no other persons are in event.

### 14.9.4.5 ui

`Ui::EventDialog*` `EventDialog::ui` [private]

User interface of the Qt framework.

The documentation for this class was generated from the following files:

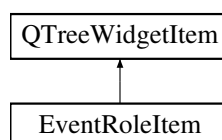
- [graphics/dialogs.h](#)
- [graphics/dialogs.cpp](#)

## 14.10 EventRoleItem Class Reference

Tree widget item for persons in events.

```
#include <graphics_items.h>
```

Inheritance diagram for EventRoleItem:



## Public Member Functions

- [EventRoleItem](#) (const std::string &role, size\_t id=0)
- size\_t [getId](#) ()
- void [setPerson](#) (const QString &name, size\_t id=0)

## Private Attributes

- size\_t [id\\_](#)  
*Id of the person.*

### 14.10.1 Detailed Description

Tree widget item for persons in events.

### 14.10.2 Constructor & Destructor Documentation

#### 14.10.2.1 EventRoleItem()

```
EventRoleItem::EventRoleItem (  
    const std::string & role,  
    size_t id = 0 ) [explicit]
```

Default constructor.

#### Parameters

|             |                               |
|-------------|-------------------------------|
| <i>role</i> | String representing the role. |
| <i>id</i>   | Id of the person if any.      |

### 14.10.3 Member Function Documentation

#### 14.10.3.1 getId()

```
size_t EventRoleItem::getId ( )
```

Get the id of the person.

#### Returns

Id of the person.

#### 14.10.3.2 setPerson()

```
void EventRoleItem::setPerson (  
    const QString & name,  
    size_t id = 0 )
```

Set person to the role.

## Parameters

|             |                     |
|-------------|---------------------|
| <i>name</i> | Name of the person. |
| <i>id</i>   | Id of the person.   |

## 14.10.4 Member Data Documentation

### 14.10.4.1 id\_

```
size_t EventRoleItem::id_ [private]
```

Id of the person.

The documentation for this class was generated from the following files:

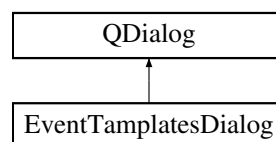
- [graphics/graphics\\_items.h](#)
- [graphics/graphics\\_items.cpp](#)

## 14.11 EventTemplatesDialog Class Reference

Dialog showing all event templates and possibly editing them or creating new ones.

```
#include <dialogs.h>
```

Inheritance diagram for EventTemplatesDialog:



### Public Slots

- void [addDefaults](#) ()  
*Add default templates.*
- void [addNewRole](#) ()  
*Add new role to the list.*
- void [addNewTemplate](#) ()  
*Show empty boxes to add new template.*
- void [removeRole](#) ()  
*Remove role from the list.*
- void [removeTemplate](#) ()  
*Remove template from the list.*
- void [saveTemplate](#) ()  
*Save template or create a new one.*
- void [showTemplate](#) (QTreeWidgetItem \*item)

## Public Member Functions

- [EventTemplatesDialog](#) ([FamilyTree](#) \*FT, [QWidget](#) \*parent=nullptr)
- [~EventTemplatesDialog](#) ()

*Default destructor.*

## Private Member Functions

- void [clear](#) ()  
*Clear all boxes.*
- void [refreshListWidget](#) ()  
*Refresh list of all relation templates.*
- void [saveTemplate](#) ([EventTemplate](#) \*event)
- void [setEnabledWidgets](#) (bool enabled)

## Private Attributes

- bool [addingNew\\_](#)  
*When we are adding new template.*
- [FamilyTree](#) \* [FT\\_](#)  
*Pointer to the family tree.*
- [Ui::EventTemplatesDialog](#) \* [ui](#)  
*User interface of the Qt framework.*

### 14.11.1 Detailed Description

Dialog showing all event templates and possibly editing them or creating new ones.

### 14.11.2 Constructor & Destructor Documentation

#### 14.11.2.1 EventTemplatesDialog()

```
EventTemplatesDialog::EventTemplatesDialog (
    FamilyTree * FT,
    QWidget * parent = nullptr ) [explicit]
```

Default constructor.

#### Parameters

|               |                             |
|---------------|-----------------------------|
| <i>FT</i>     | Pointer to the family tree. |
| <i>parent</i> | The Qt Widget parent.       |

#### 14.11.2.2 ~EventTemplatesDialog()

```
EventTemplatesDialog::~EventTemplatesDialog ( )
```

Default destructor.

### 14.11.3 Member Function Documentation

#### 14.11.3.1 addDefaults

```
void EventTemplatesDialog::addDefaults ( ) [slot]
```

Add default templates.

#### 14.11.3.2 addNewRole

```
void EventTemplatesDialog::addNewRole ( ) [slot]
```

Add new role to the list.

#### 14.11.3.3 addNewTemplate

```
void EventTemplatesDialog::addNewTemplate ( ) [slot]
```

Show empty boxes to add new template.

#### 14.11.3.4 clear()

```
void EventTemplatesDialog::clear ( ) [private]
```

Clear all boxes.

#### 14.11.3.5 refreshListWidget()

```
void EventTemplatesDialog::refreshListWidget ( ) [private]
```

Refresh list of all relation templates.

#### 14.11.3.6 removeRole

```
void EventTemplatesDialog::removeRole ( ) [slot]
```

Remove role from the list.

#### 14.11.3.7 removeTemplate

```
void EventTemplatesDialog::removeTemplate ( ) [slot]
```

Remove template from the list.

#### 14.11.3.8 saveTemplate [1/2]

```
void EventTemplatesDialog::saveTemplate ( ) [slot]
```

Save template or create a new one.

#### 14.11.3.9 saveTemplate() [2/2]

```
void EventTemplatesDialog::saveTemplate (
    EventTemplate * event ) [private]
```

Overwrite data of the event template by the input from ui.

## Parameters

|              |   |
|--------------|---|
| <i>event</i> | Pointer to which template to overwrite. |
|--------------|---|

**14.11.3.10 setEnabledWidgets()**

```
void EventTemplatesDialog::setEnabledWidgets (
    bool enabled ) [private]
```

Set enabled for all widgets.

## Parameters

|                |  |
|----------------|--|
| <i>enabled</i> | If the widgets should be enabled or not. |
|----------------|--|

**14.11.3.11 showTemplate**

```
void EventTemplatesDialog::showTemplate (
    QTreeWidgetItem * item ) [slot]
```

Show template details.

## Parameters

|             |                        |
|-------------|------------------------|
| <i>item</i> | Which item is clicked. |
|-------------|------------------------|

**14.11.4 Member Data Documentation****14.11.4.1 addingNew\_**

```
bool EventTemplatesDialog::addingNew_ [private]
```

When we are adding new template.

**14.11.4.2 FT\_**

```
FamilyTree* EventTemplatesDialog::FT_ [private]
```

Pointer to the family tree.

**14.11.4.3 ui**

```
Ui::EventTemplatesDialog* EventTemplatesDialog::ui [private]
```

User interface of the Qt framework.

The documentation for this class was generated from the following files:

- [graphics/dialogs.h](#)
- [graphics/dialogs.cpp](#)

## 14.12 EventTemplate Class Reference

Template for events. It specifies whether the event includes a date, a place or if more persons are involved and if they have some special roles.

```
#include <config.h>
```

### Public Member Functions

- [EventTemplate](#) ()  
*Default constructor.*
- bool [containsDate](#) () const
- bool [containsPlace](#) () const
- const std::string & [getDescription](#) () const
- size\_t [getId](#) () const
- const std::vector< std::string > & [getRoles](#) () const
- const std::string & [getTitle](#) () const
- bool [hasMorePeopleInvolved](#) () const
- bool [hasRoles](#) () const
- void [readJson](#) (const Json::Value &value)
- void [setContainsDate](#) (bool hasDate)
- void [setContainsPlace](#) (bool hasPlace)
- std::string & [setDescription](#) ()
- void [setId](#) (size\_t id)
- void [setMorePeopleInvolved](#) (bool morePeople)
- void [setRoles](#) (std::vector< std::string > &roles)
- std::string & [setTitle](#) ()
- std::string [str](#) () const

### Private Attributes

- bool [containsDate\\_](#)  
*If the event contains a date or not.*
- bool [containsPlace\\_](#)  
*If the event contains a place or not.*
- std::string [description\\_](#)  
*Description of the event.*
- size\_t [id\\_](#)  
*Special ID of this template. May change during the execution.*
- bool [morePeopleInvolved\\_](#)  
*If more people are involved. Otherwise only one is involved.*
- std::vector< std::string > [roles\\_](#)  
*Vector of special role names.*
- std::string [title\\_](#)  
*Title of the event.*

### 14.12.1 Detailed Description

Template for events. It specifies whether the event includes a date, a place or if more persons are involved and if they have some special roles.

## 14.12.2 Constructor & Destructor Documentation

### 14.12.2.1 EventTemplate()

```
EventTemplate::EventTemplate ( )
```

Default constructor.

## 14.12.3 Member Function Documentation

### 14.12.3.1 containsDate()

```
bool EventTemplate::containsDate ( ) const
```

If the event contains a date or not.

#### Returns

True if it contains a date, false otherwise.

### 14.12.3.2 containsPlace()

```
bool EventTemplate::containsPlace ( ) const
```

If the event contains a place or not.

#### Returns

True if the event contains a place, false otherwise.

### 14.12.3.3 getDescription()

```
const std::string & EventTemplate::getDescription ( ) const
```

Get the description of the event.

#### Returns

Constant reference to the description.

### 14.12.3.4 getId()

```
size_t EventTemplate::getId ( ) const
```

Get the ID of this event template.

#### Returns

ID of this event template.



### 14.12.3.5 getRoles()

```
const std::vector< std::string > & EventTemplate::getRoles ( ) const
```

Get the vector of special roles.

#### Returns

Constant reference to the vector of role names.

### 14.12.3.6 getTitle()

```
const std::string & EventTemplate::getTitle ( ) const
```

Get the title of this event.

#### Returns

Constant reference to the title.

### 14.12.3.7 hasMorePeopleInvolved()

```
bool EventTemplate::hasMorePeopleInvolved ( ) const
```

If there is more people involved in this event.

#### Returns

True if more people are involved.

### 14.12.3.8 hasRoles()

```
bool EventTemplate::hasRoles ( ) const
```

If the event has special roles.

#### Returns

True if special roles were given.

### 14.12.3.9 readJson()

```
void EventTemplate::readJson (
    const Json::Value & value )
```

Read the data from JSON.

**Parameters**

|              |                   |
|--------------|-------------------|
| <i>value</i> | Given JSON value. |
|--------------|-------------------|

**14.12.3.10 setContainsDate()**

```
void EventTemplate::setContainsDate (
    bool hasDate )
```

Set if the template contains date or not.

**Parameters**

|                |                                  |
|----------------|----------------------------------|
| <i>hasDate</i> | If the template contains a date. |
|----------------|----------------------------------|

**14.12.3.11 setContainsPlace()**

```
void EventTemplate::setContainsPlace (
    bool hasPlace )
```

Set if the template contains place or not.

**Parameters**

|                 |                                   |
|-----------------|-----------------------------------|
| <i>hasPlace</i> | If the template contains a place. |
|-----------------|-----------------------------------|

**14.12.3.12 setDescription()**

```
std::string & EventTemplate::setDescription ( )
```

Set the description of the event by giving the reference to the string.

**Returns**

Reference to the description.

**14.12.3.13 setId()**

```
void EventTemplate::setId (
    size_t id )
```

Set the ID of this event template.

**Parameters**

|           |         |
|-----------|---------|
| <i>id</i> | New ID. |
|-----------|---------|

#### 14.12.3.14 setMorePeopleInvolved()

```
void EventTemplate::setMorePeopleInvolved (
    bool morePeople )
```

Set if more people are involved in this event.

##### Parameters

|                   |                                   |
|-------------------|-----------------------------------|
| <i>morePeople</i> | True if more people are involved. |
|-------------------|-----------------------------------|

#### 14.12.3.15 setRoles()

```
void EventTemplate::setRoles (
    std::vector< std::string > & roles )
```

Set the roles.

##### Parameters

|              |                           |
|--------------|---------------------------|
| <i>roles</i> | Vector of all role names. |
|--------------|---------------------------|

#### 14.12.3.16 setTitle()

```
std::string & EventTemplate::setTitle ( )
```

Set the title of the event by giving the reference to the string.

##### Returns

Reference to the title.

#### 14.12.3.17 str()

```
std::string EventTemplate::str ( ) const
```

Represent event template as a string.

##### Returns

Newly constructed string representing the template.

### 14.12.4 Member Data Documentation

#### 14.12.4.1 containsDate\_

```
bool EventTemplate::containsDate_ [private]
```

If the event contains a date or not.

#### 14.12.4.2 containsPlace\_

```
bool EventTemplate::containsPlace_ [private]
```

If the event contains a place or not.

#### 14.12.4.3 description\_

```
std::string EventTemplate::description_ [private]
```

Description of the event.

#### 14.12.4.4 id\_

```
size_t EventTemplate::id_ [private]
```

Special ID of this template. May change during the execution.

#### 14.12.4.5 morePeopleInvolved\_

```
bool EventTemplate::morePeopleInvolved_ [private]
```

If more people are involved. Otherwise only one is involved.

#### 14.12.4.6 roles\_

```
std::vector<std::string> EventTemplate::roles_ [private]
```

Vector of special role names.

#### 14.12.4.7 title\_

```
std::string EventTemplate::title_ [private]
```

Title of the event.

The documentation for this class was generated from the following files:

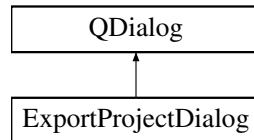
- [core/config.h](#)
- [core/config.cpp](#)

## 14.13 ExportProjectDialog Class Reference

Dialog for exporting project.

```
#include <dialogs.h>
```

Inheritance diagram for ExportProjectDialog:



### Public Slots

- void [checkPerson](#) ()  
*Let the person check box be always checked.*
- void [exportProject](#) ()  
*Export project upon clicking OK.*

### Public Member Functions

- [ExportProjectDialog](#) ([FamilyTree](#) \*FT, QWidget \*parent=nullptr)
- [~ExportProjectDialog](#) ()  
*Default destructor.*

### Private Attributes

- [FamilyTree](#) \* FT\_  
*Pointer to the family tree.*
- Ui::ExportProjectDialog \* ui  
*Used Qt user interface.*

### 14.13.1 Detailed Description

Dialog for exporting project.

### 14.13.2 Constructor & Destructor Documentation

#### 14.13.2.1 ExportProjectDialog()

```
ExportProjectDialog::ExportProjectDialog (
    FamilyTree * FT,
    QWidget * parent = nullptr ) [explicit]
```

Default constructor.

## Parameters

|               |                                  |
|---------------|----------------------------------|
| <i>FT</i>     | Pointer to the used Family tree. |
| <i>parent</i> | The Qt Widget parent.            |

**14.13.2.2 ~ExportProjectDialog()**

```
ExportProjectDialog::~ExportProjectDialog ( )
```

Default destructor.

**14.13.3 Member Function Documentation****14.13.3.1 checkPerson**

```
void ExportProjectDialog::checkPerson ( ) [slot]
```

Let the person check box be always checked.

**14.13.3.2 exportProject**

```
void ExportProjectDialog::exportProject ( ) [slot]
```

Export project upon clicking OK.

**14.13.4 Member Data Documentation****14.13.4.1 FT\_**

```
FamilyTree* ExportProjectDialog::FT_ [private]
```

Pointer to the family tree.

**14.13.4.2 ui**

```
Ui::ExportProjectDialog* ExportProjectDialog::ui [private]
```

Used Qt user interface.

The documentation for this class was generated from the following files:

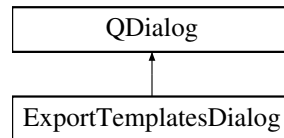
- [graphics/dialogs.h](#)
- [graphics/dialogs.cpp](#)

## 14.14 ExportTemplatesDialog Class Reference

Dialog for exporting some of the event and relation templates.

```
#include <dialogs.h>
```

Inheritance diagram for ExportTemplatesDialog:



### Public Slots

- void [exportTemplates](#) ()  
*When OK button was pressed load all selected templates and export them.*

### Public Member Functions

- [ExportTemplatesDialog](#) ([FamilyTree](#) \*FT, QWidget \*parent=nullptr)
- [~ExportTemplatesDialog](#) ()  
*Default destructor.*

### Private Attributes

- [FamilyTree](#) \* FT\_  
*Pointer to the family tree.*
- Ui::ExportTemplatesDialog \* ui  
*Used Qt user interface.*

### 14.14.1 Detailed Description

Dialog for exporting some of the event and relation templates.

### 14.14.2 Constructor & Destructor Documentation

#### 14.14.2.1 ExportTemplatesDialog()

```
ExportTemplatesDialog::ExportTemplatesDialog (
    FamilyTree * FT,
    QWidget * parent = nullptr ) [explicit]
```

Default constructor.

## Parameters

|               |                             |
|---------------|-----------------------------|
| <i>FT</i>     | Pointer to the family tree. |
| <i>parent</i> | The Qt Widget parent.       |

**14.14.2.2 ~ExportTemplatesDialog()**

```
ExportTemplatesDialog::~ExportTemplatesDialog ( )
```

Default destructor.

**14.14.3 Member Function Documentation****14.14.3.1 exportTemplates**

```
void ExportTemplatesDialog::exportTemplates ( ) [slot]
```

When OK button was pressed load all selected templates and export them.

**14.14.4 Member Data Documentation****14.14.4.1 FT\_**

```
FamilyTree* ExportTemplatesDialog::FT_ [private]
```

Pointer to the family tree.

**14.14.4.2 ui**

```
Ui::ExportTemplatesDialog* ExportTemplatesDialog::ui [private]
```

Used Qt user interface.

The documentation for this class was generated from the following files:

- [graphics/dialogs.h](#)
- [graphics/dialogs.cpp](#)

**14.15 FamilyTree Class Reference**

Main class for holding all data of a family tree. Also it acts as a bridge to the core of the application.

```
#include <family_tree.h>
```



## Public Member Functions

- [FamilyTree](#) ()  
*Default constructor.*
- [Event](#) \* [addEvent](#) ()
- [Event](#) \* [addEvent](#) (size\_t index)
- [File](#) \* [addFile](#) (const std::string &name, [FileType](#) type=[GENERAL\\_FILE](#))
- [File](#) \* [addFile](#) (const std::string &name, size\_t index, [FileType](#) type=[GENERAL\\_FILE](#))
- [Person](#) \* [addPerson](#) ()
- [Person](#) \* [addPerson](#) (size\_t index)
- [Relation](#) \* [addRelation](#) ()
- [Relation](#) \* [addRelation](#) (size\_t index)
- void [append](#) ([FamilyTree](#) &other)
- std::pair< bool, std::string > [checkFileConsistence](#) ()
- void [clear](#) ()  
*Clear all data and set it for a new project.*
- void [clearProjectPaths](#) ()  
*Clear all project paths except for this one.*
- size\_t [copyFile](#) (const std::string &filePath, [FileType](#) type=[GENERAL\\_FILE](#))
- void [createDatabase](#) (const std::string &dirPath)
- size\_t [createEmptyFile](#) (const std::string &name, [FileType](#) type=[GENERAL\\_FILE](#))
- void [enableHelpOnStartup](#) (bool help)
- void [exportProject](#) (const std::string &filePath, bool includingEvents, bool includingRelations)
- std::string [createResourcesDir](#) (const std::string &target)
- void [exportTemplates](#) (const std::vector< size\_t > &eventTemplates, const std::vector< size\_t > &relations, const std::string &filename)
- std::optional< [Event](#) \* > [getEvent](#) (size\_t id)
- const std::map< size\_t, std::unique\_ptr< [Event](#) > > & [getEvents](#) () const
- std::optional< [File](#) \* > [getFile](#) (size\_t id, [FileType](#) type=[GENERAL\\_FILE](#))
- void [getFileAbsolutePath](#) (std::string &absolutePath, size\_t id, [FileType](#) type=[GENERAL\\_FILE](#))
- const std::map< size\_t, std::unique\_ptr< [File](#) > > & [getFiles](#) ([FileType](#) type=[GENERAL\\_FILE](#))
- std::vector< std::string > [getOrphanFiles](#) ()
- [Person](#) \* [getMainPerson](#) ()
- const std::map< size\_t, std::unique\_ptr< [Person](#) > > & [getPersons](#) ()
- std::optional< [Person](#) \* > [getPerson](#) (size\_t id)
- const std::vector< std::string > & [getProjectPaths](#) () const
- std::optional< [Relation](#) \* > [getRelation](#) (size\_t id)
- const std::map< size\_t, std::unique\_ptr< [Relation](#) > > & [getRelations](#) ()
- std::vector< [RelationSuggestion](#) > [getRelationSuggestions](#) (size\_t relId)
- [Settings](#) \* [getSettings](#) ()
- void [getUriFile](#) (size\_t id, std::string &filePath, [FileType](#) type=[GENERAL\\_FILE](#))
- bool [importProject](#) (const std::string &filePath)
- bool [importRootDir](#) (const std::string &dirPath)
- std::pair< bool, size\_t > [importTemplates](#) (const std::string &filePath)
- bool [isDirectorySet](#) ()
- bool [isSaved](#) () const
- void [log](#) (const std::string &error)
- std::pair< bool, bool > [openDatabase](#) (const std::string &dirPath, std::string &errorMessage, std::string &backupFile)
- bool [openDatabase](#) (std::string &errorMessage)
- bool [openSingleFileDatabase](#) (const std::string &filePath)
- void [printHtml](#) (const std::string &dirPath)
- void [printMainPersonHtml](#) (const std::string &outputFile)
- void [promoteRelation](#) (size\_t id)

- void `removeBackup` ()  
*Remove backup file if there is any.*
- void `removeEvent` (size\_t id)
- void `removeEventTemplate` (size\_t id)
- size\_t `removeOrphanFiles` ()
- void `removePerson` ()  
*Remove main person from the tree. Also delete all of his appearances.*
- void `removeRelation` (size\_t id)
- void `removeRelationTemplate` (size\_t id)
- void `renameFile` (size\_t id, const std::string &newFilename, `FileType` type)
- bool `restoreBackup` (const std::string &backupFile, std::string &errorMessage)
- void `setMainPerson` (`Person` \*p)
- void `setMainPerson` (size\_t id)
- void `setRelation` (size\_t relId, size\_t pers1Id, size\_t pers2Id, size\_t templId)
- void `setUnsaved` ()  
*Set unsaved state.*
- bool `showHelpOnStartup` ()
- bool `storeDatabase` ()
- size\_t `templatesBasedOnEventTemplate` (size\_t templateId)
- size\_t `templatesBasedOnRelationTemplate` (size\_t templateId)
- void `updateEventsWithTemplate` (size\_t templ)
- bool `writeConfig` (const std::string &newPath)

### Private Member Functions

- bool `checkFileTypeConsistence` (std::ostream &os, `FileType` type)
- void `getGeneralOrphaFiles` (std::vector< std::string > &files, const std::map< size\_t, std::unique\_ptr< `File` > > &container, `FileType` type)
- void `getParentSuggestions` (std::vector< `RelationSuggestion` > &suggestions, const `Person` \*parent, const `Person` \*child, size\_t id, `Trait` trait, const std::set< size\_t > &forbiddenPersonParent, const std::set< size\_t > &forbiddenPersonChild)
- void `getPartnersSuggestions` (std::vector< `RelationSuggestion` > &suggestions, const `Person` \*person, const `Person` \*second, size\_t id, const std::set< size\_t > &forbiddenPersons)
- void `getSiblingsSuggestions` (std::vector< `RelationSuggestion` > &suggestions, const `Person` \*person, const `Person` \*second, size\_t id, const std::set< size\_t > &forbiddenPersons)
- bool `loadDatabaseFile` (const std::string &file, std::function< void(const Json::Value &)> reader, std::string &errorMessage)
- void `printCss` (const std::string &dirPath)
- void `printFiles` (std::ostream &os, const std::map< size\_t, std::unique\_ptr< `File` > > &container, const std::string &label)
- void `printEvents` (std::ostream &os)
- void `printHtmlPerson` (std::ostream &os, bool links, `Person` \*p)
- void `printPeople` (std::ostream &os)
- void `printRelations` (std::ostream &os)
- void `readJsonEvents` (const Json::Value &value)
- void `readJsonFiles` (const Json::Value &value, std::map< size\_t, std::unique\_ptr< `File` > > &container, const std::string &label, size\_t &index)
- void `readJsonPeople` (const Json::Value &value)
- void `readJsonRelations` (const Json::Value &value)
- size\_t `removeGeneralOrphanFile` (std::map< size\_t, std::unique\_ptr< `File` > > &container, `FileType` type, const std::string &dir)
- void `storeFileDatabase` (`FileType` type)

## Private Attributes

- `std::map< size_t, std::unique_ptr< Event > > allEvents_`  
*All the events in the tree.*
- `std::map< size_t, std::unique_ptr< File > > allFiles_`  
*All files in app.*
- `std::map< size_t, std::unique_ptr< File > > allMedia_`  
*All media in app.*
- `std::map< size_t, std::unique_ptr< File > > allNotes_`  
*All notes in app.*
- `std::map< size_t, std::unique_ptr< Person > > allPersons_`  
*Map of all persons in family tree.*
- `std::map< size_t, std::unique_ptr< Relation > > allRelations_`  
*All relations in the tree.*
- `size_t event_index_`  
*Last free index for event. Always start from 1.*
- `size_t file_index_`  
*Last free id for file. Always start from 1.*
- `Person * mainPerson_`  
*Main person showing as the centre of the tree.*
- `size_t media_index_`  
*Last free id for media. Always start from 1.*
- `size_t note_index_`  
*Last free id for note. Always start from 1.*
- `bool openHelp`  
*If help window should be shown.*
- `Parser parser_`  
*[Parser](#) for working with locally stored database.*
- `size_t person_index_`  
*Last free id for person. Always start from 1.*
- `std::vector< std::string > projectPaths`  
*Paths to existing projects.*
- `size_t relation_index_`  
*Last free index for relation.*
- `bool savedChanges_`  
*If all changes were saved or not.*
- `Settings settings_`  
*[Settings](#) of the app.*

### 14.15.1 Detailed Description

Main class for holding all data of a family tree. Also it acts as a bridge to the core of the application.

### 14.15.2 Constructor & Destructor Documentation

#### 14.15.2.1 FamilyTree()

```
FamilyTree::FamilyTree ( )
```

Default constructor.

### 14.15.3 Member Function Documentation

#### 14.15.3.1 addEvent() [1/2]

```
Event * FamilyTree::addEvent ( )
```

Add new event to the family tree.

##### Returns

Pointer to the new event.

#### 14.15.3.2 addEvent() [2/2]

```
Event * FamilyTree::addEvent (
    size_t index )
```

Add new event to the family tree with known index.

##### Parameters

|              |              |
|--------------|--------------|
| <i>index</i> | Known index. |
|--------------|--------------|

##### Returns

Pointer to the new event.

#### 14.15.3.3 addFile() [1/2]

```
File * FamilyTree::addFile (
    const std::string & name,
    FileType type = GENERAL_FILE )
```

Add file to the database.

##### Parameters

|             |   |
|-------------|---|
| <i>name</i> | Given name of the file.                     |
| <i>type</i> | If the file is general file, media or note. |

##### Returns

Pointer to the new file.

#### 14.15.3.4 addFile() [2/2]

```
File * FamilyTree::addFile (
    const std::string & name,
```

```
size_t index,  
FileType type = GENERAL_FILE )
```

Add file to the database with known index.

#### Parameters

|              |   |
|--------------|---|
| <i>name</i>  | Given name of the file.                     |
| <i>index</i> | Known index.                                |
| <i>type</i>  | If the file is general file, media or note. |

#### Returns

Pointer to the new file.

#### 14.15.3.5 addPerson() [1/2]

```
Person * FamilyTree::addPerson ( )
```

Add person to the database.

#### Returns

Pointer to that new person.

#### 14.15.3.6 addPerson() [2/2]

```
Person * FamilyTree::addPerson (   
    size_t index )
```

Add person to the database with known index.

#### Parameters

|              |              |
|--------------|--------------|
| <i>index</i> | Known index. |
|--------------|--------------|

#### Returns

Pointer to that new person.

#### 14.15.3.7 addRelation() [1/2]

```
Relation * FamilyTree::addRelation ( )
```

Add relation to the family tree.

#### Returns

Pointer to the new relation.

#### 14.15.3.8 addRelation() [2/2]

```
Relation * FamilyTree::addRelation (
    size_t index )
```

Add relation to the family tree with known index.

##### Parameters

|              |              |
|--------------|--------------|
| <i>index</i> | Known index. |
|--------------|--------------|

##### Returns

Pointer to the new relation.

#### 14.15.3.9 append()

```
void FamilyTree::append (
    FamilyTree & other )
```

Append the whole project by another one. Note that the second family tree will be corrupted. Because all the indexes must be changed. Though the second tree won't be saved.

##### Parameters

|              |                     |
|--------------|---------------------|
| <i>other</i> | Second Family tree. |
|--------------|---------------------|

#### 14.15.3.10 checkFileConsistence()

```
std::pair< bool, std::string > FamilyTree::checkFileConsistence ( )
```

Check the consistence of stored file.

##### Returns

Bool if it is consistent and written details of all wrong files.

#### 14.15.3.11 checkFileTypeConsistence()

```
bool FamilyTree::checkFileTypeConsistence (
    std::ostream & os,
    FileType type ) [private]
```

Check file consistence of a single type.

##### Parameters

|             |                         |
|-------------|-------------------------|
| <i>os</i>   | Where to write error.   |
| <i>type</i> | Which type to look for. |

#### 14.15.3.12 clear()

```
void FamilyTree::clear ( )
```

Clear all data and set it for a new project.

#### 14.15.3.13 clearProjectPaths()

```
void FamilyTree::clearProjectPaths ( )
```

Clear all project paths except for this one.

#### 14.15.3.14 copyFile()

```
size_t FamilyTree::copyFile (
    const std::string & filePath,
    FileType type = GENERAL_FILE )
```

Copy an existing file.

##### Parameters

|                 |  |
|-----------------|--|
| <i>filePath</i> | Path to the original file on the disk. |
| <i>type</i>     | What type of file it is.               |

##### Returns

Id of the new file.

#### 14.15.3.15 createDatabase()

```
void FamilyTree::createDatabase (
    const std::string & dirPath )
```

Create a new empty database.

##### Parameters

|                |                             |
|----------------|-----------------------------|
| <i>dirPath</i> | Path to the root directory. |
|----------------|-----------------------------|

#### 14.15.3.16 createEmptyFile()

```
size_t FamilyTree::createEmptyFile (
    const std::string & name,
    FileType type = GENERAL_FILE )
```

Create an empty file.

**Parameters**

|             |  |
|-------------|--|
| <i>name</i> | Name of the file. Not the real name on disk. |
| <i>type</i> | What type of file it is.                     |

**Returns**

Id of the new file.

**14.15.3.17 createResourcesDir()**

```
std::string FamilyTree::createResourcesDir (  
    const std::string & target )
```

Create resource file for diagrams for HTML export.

**Parameters**

|               |  |
|---------------|--|
| <i>target</i> | Where the resource dir should be (either a path to the directory or a file). |
|---------------|--|

**Returns**

String representing the path to the resources directory.

**14.15.3.18 enableHelpOnStartup()**

```
void FamilyTree::enableHelpOnStartup (  
    bool help )
```

Switch if the help window should be opened at start up, also write it to configuration.

**Parameters**

|             |                                |
|-------------|--------------------------------|
| <i>help</i> | If it should be opened or not. |
|-------------|--------------------------------|

**14.15.3.19 exportProject()**

```
void FamilyTree::exportProject (  
    const std::string & filePath,  
    bool includingEvents,  
    bool includingRelations )
```

Export this project in one single JSON file.

**Parameters**

|                           |   |
|---------------------------|---|
| <i>filePath</i>           | Path to the export file.                |
| <i>includingEvents</i>    | If the export should include events.    |
| <i>includingRelations</i> | If the export should include relations. |



**14.15.3.20 exportTemplates()**

```
void FamilyTree::exportTemplates (
    const std::vector< size_t > & eventTemplates,
    const std::vector< size_t > & relTemplates,
    const std::string & filename )
```

Export given templates to an output file in JSON format.

**Parameters**

|                       |                               |
|-----------------------|-------------------------------|
| <i>eventTemplates</i> | Vector of event templates.    |
| <i>relTemplates</i>   | Vector of relation templates. |
| <i>filename</i>       | Path to the output file.      |

**14.15.3.21 getEvent()**

```
std::optional< Event * > FamilyTree::getEvent (
    size_t id )
```

Get pointer to the event.

**Parameters**

|           |                  |
|-----------|------------------|
| <i>id</i> | Id of the event. |
|-----------|------------------|

**Returns**

Get optionally pointer to the event or empty.

**14.15.3.22 getEvents()**

```
const std::map< size_t, std::unique_ptr< Event > > & FamilyTree::getEvents ( ) const
```

Get constant reference to the container of all events.

**Returns**

Constant reference to the map of all events.

**14.15.3.23 getFile()**

```
std::optional< File * > FamilyTree::getFile (
    size_t id,
    FileType type = GENERAL_FILE )
```

Get pointer to given file.

**Parameters**

|             |                           |
|-------------|---------------------------|
| <i>id</i>   | Id of the file.           |
| <i>type</i> | What type of file to get. |

**Returns**

Optionally pointer to the file or empty.

**14.15.3.24 getFileAbsolutePath()**

```
void FamilyTree::getFileAbsolutePath (
    std::string & absolutePath,
    size_t id,
    FileType type = GENERAL_FILE )
```

Get the absolute path to the given file.

**Parameters**

|                     |  |
|---------------------|--|
| <i>absolutePath</i> | Where will be the absolute paths stored. |
| <i>id</i>           | Id of the file.                          |
| <i>type</i>         | Which type is used.                      |

**14.15.3.25 getFiles()**

```
const std::map< size_t, std::unique_ptr< File > > & FamilyTree::getFiles (
    FileType type = GENERAL_FILE )
```

Get constant reference to the vector of all files.

**Parameters**

|             |                                    |
|-------------|------------------------------------|
| <i>type</i> | What type of files the vector has. |
|-------------|------------------------------------|

**Returns**

Constant reference to the map of files with given type.

**14.15.3.26 getGeneralOrphaFiles()**

```
void FamilyTree::getGeneralOrphaFiles (
    std::vector< std::string > & files,
    const std::map< size_t, std::unique_ptr< File > > & container,
    FileType type ) [private]
```

Get all files that would be removed.

## Parameters

|                  |                             |
|------------------|-----------------------------|
| <i>files</i>     | Where to store these files. |
| <i>container</i> | Which map to use.           |
| <i>type</i>      | Which type we are using.    |

**14.15.3.27 getMainPerson()**

```
Person * FamilyTree::getMainPerson ( )
```

Get the main person.

## Returns

Pointer to the main person.

**14.15.3.28 getOrphanFiles()**

```
std::vector< std::string > FamilyTree::getOrphanFiles ( )
```

Get all the orphan files.

## Returns

Vector of all files, that will be permanently deleted.

**14.15.3.29 getParentSuggestions()**

```
void FamilyTree::getParentSuggestions (
    std::vector< RelationSuggestion > & suggestions,
    const Person * parent,
    const Person * child,
    size_t id,
    Trait trait,
    const std::set< size_t > & forbiddenPersonParent,
    const std::set< size_t > & forbiddenPersonChild ) [private]
```

Append relation suggestions by a parent relations.

## Parameters

|                              |   |
|------------------------------|---|
| <i>suggestions</i>           | Container of all suggestions.                         |
| <i>parent</i>                | Which person is parent.                               |
| <i>child</i>                 | Which person is child.                                |
| <i>id</i>                    | Id of this relation.                                  |
| <i>trait</i>                 | Trait of this relation.                               |
| <i>forbiddenPersonParent</i> | Which persons are forbidden to bind to with a parent. |
| <i>forbiddenPersonChild</i>  | Which persons are forbidden to bind to with a child.  |

**14.15.3.30 getPartnersSuggestions()**

```
void FamilyTree::getPartnersSuggestions (
    std::vector< RelationSuggestion > & suggestions,
    const Person * person,
    const Person * second,
    size_t id,
    const std::set< size_t > & forbiddenPersons ) [private]
```

Append relation suggestions by partners relations.

**Parameters**

|                         |  |
|-------------------------|--|
| <i>suggestions</i>      | Container of all suggestions.                          |
| <i>person</i>           | Which person is the main one.                          |
| <i>second</i>           | Which person is the second one.                        |
| <i>id</i>               | Id of this relation.                                   |
| <i>forbiddenPersons</i> | Which persons are forbidden to bind to a given person. |

**14.15.3.31 getPerson()**

```
std::optional< Person * > FamilyTree::getPerson (
    size_t id )
```

Get pointer to the person by its id.

**Parameters**

|           |                   |
|-----------|-------------------|
| <i>id</i> | Id of the person. |
|-----------|-------------------|

**Returns**

Optionally pointer to the person with its id or empty.

**14.15.3.32 getPersons()**

```
const std::map< size_t, std::unique_ptr< Person > > & FamilyTree::getPersons ( )
```

Get the container of persons.

**Returns**

Constant reference to the container.

**14.15.3.33 getProjectPaths()**

```
const std::vector< std::string > & FamilyTree::getProjectPaths ( ) const
```

Get the constant reference to the paths to the roots of projects.

**Returns**

Constant reference to the vector of all strings representing root directories.

### 14.15.3.34 getRelation()

```
std::optional< Relation * > FamilyTree::getRelation (
    size_t id )
```

Get pointer to the relation.

#### Parameters

|           |                     |
|-----------|---------------------|
| <i>id</i> | Id of the relation. |
|-----------|---------------------|

#### Returns

Optionally pointer to the relation or empty.

### 14.15.3.35 getRelations()

```
const std::map< size_t, std::unique_ptr< Relation > > & FamilyTree::getRelations ( )
```

Get the constant reference to the map of all relations.

#### Returns

Constant reference to the map of all relations.

### 14.15.3.36 getRelationSuggestions()

```
std::vector< RelationSuggestion > FamilyTree::getRelationSuggestions (
    size_t relId )
```

Get relation suggestions to the given relation.

#### Parameters

|              |                           |
|--------------|---------------------------|
| <i>relId</i> | Id of the given relation. |
|--------------|---------------------------|

#### Returns

vector of all suggestions.

### 14.15.3.37 getSettings()

```
Settings * FamilyTree::getSettings ( )
```

Get pointer to the settings.

#### Returns

Pointer to the settings.

#### 14.15.3.38 getSiblingsSuggestions()

```
void FamilyTree::getSiblingsSuggestions (
    std::vector< RelationSuggestion > & suggestions,
    const Person * person,
    const Person * second,
    size_t id,
    const std::set< size_t > & forbiddenPersons ) [private]
```

Append relation suggestions by siblings relations.

##### Parameters

|                         |  |
|-------------------------|--|
| <i>suggestions</i>      | Container of all suggestions.                          |
| <i>person</i>           | Which person is the main one.                          |
| <i>second</i>           | Which person is the second one.                        |
| <i>id</i>               | Id of this relation. – This will be changed            |
| <i>forbiddenPersons</i> | Which persons are forbidden to bind to a given person. |

#### 14.15.3.39 getUrlFile()

```
void FamilyTree::getUrlFile (
    size_t id,
    std::string & fileUrl,
    FileType type = GENERAL_FILE )
```

Get string representing URL to local file.

##### Parameters

|                |   |
|----------------|---|
| <i>id</i>      | Index of the given file.                    |
| <i>fileUrl</i> | URL of the file.                            |
| <i>type</i>    | If the file is general file, media or note. |

#### 14.15.3.40 importProject()

```
bool FamilyTree::importProject (
    const std::string & filePath )
```

Import project.

##### Parameters

|                 |                   |
|-----------------|-------------------|
| <i>filePath</i> | Path to the file. |
|-----------------|-------------------|

##### Returns

If the import was successful or not.

#### 14.15.3.41 importRootDir()

```
bool FamilyTree::importRootDir (
    const std::string & dirPath )
```

Import project from root directory.

##### Parameters

|                |                             |
|----------------|-----------------------------|
| <i>dirPath</i> | Path to the root directory. |
|----------------|-----------------------------|

##### Returns

If the import was successful or not.

#### 14.15.3.42 importTemplates()

```
std::pair< bool, size_t > FamilyTree::importTemplates (
    const std::string & filePath )
```

Import templates.

##### Parameters

|                 |                   |
|-----------------|-------------------|
| <i>filePath</i> | Path to the file. |
|-----------------|-------------------|

##### Returns

If the import was successful or not and number of imported templates.

#### 14.15.3.43 isDirectorySet()

```
bool FamilyTree::isDirectorySet ( )
```

If the root directory for the database was already set.

##### Returns

True if the directory was set.

#### 14.15.3.44 isSaved()

```
bool FamilyTree::isSaved ( ) const
```

If all changes are saved.

##### Returns

True if all changes are saved, otherwise false.

#### 14.15.3.45 loadDatabaseFile()

```
bool FamilyTree::loadDatabaseFile (
    const std::string & file,
    std::function< void(const Json::Value &)> reader,
    std::string & errorMessage ) [private]
```

Load one single file from the database.

##### Parameters

|                     |   |
|---------------------|---|
| <i>file</i>         | Which file in database is being loaded.     |
| <i>reader</i>       | Which function read the data and load them. |
| <i>errorMessage</i> | Where will the error message stored.        |

##### Returns

If the parsing was successful or not.

#### 14.15.3.46 log()

```
void FamilyTree::log (
    const std::string & error )
```

Log error code to the parser.

##### Parameters

|              |                                |
|--------------|--------------------------------|
| <i>error</i> | What is the text of the error. |
|--------------|--------------------------------|

#### 14.15.3.47 openDatabase() [1/2]

```
std::pair< bool, bool > FamilyTree::openDatabase (
    const std::string & dirPath,
    std::string & errorMessage,
    std::string & backupFile )
```

Open a database from its root directory.

##### Parameters

|                     |                                       |
|---------------------|---------------------------------------|
| <i>dirPath</i>      | Path to the root directory.           |
| <i>errorMessage</i> | To show what was the potential error. |
| <i>backupFile</i>   | Which file has a backup.              |

##### Returns

First if the parsing was successful. Second if there exists backup.



**14.15.3.48 openDatabase() [2/2]**

```
bool FamilyTree::openDatabase (
    std::string & errorMessage )
```

Read the database without setting dir path.

**Parameters**

|                     |                                  |
|---------------------|----------------------------------|
| <i>errorMessage</i> | If error occurred show it there. |
|---------------------|----------------------------------|

**Returns**

If the parsing of the database was successful or not.

**14.15.3.49 openSingleFileDatabase()**

```
bool FamilyTree::openSingleFileDatabase (
    const std::string & filePath )
```

Open database from a single file.

**Parameters**

|                 |                   |
|-----------------|-------------------|
| <i>filePath</i> | Path to the file. |
|-----------------|-------------------|

**Returns**

If the parsing was successful or not.

**14.15.3.50 printCss()**

```
void FamilyTree::printCss (
    const std::string & dirPath ) [private]
```

Print custom CSS.

**Parameters**

|                |                                    |
|----------------|------------------------------------|
| <i>dirPath</i> | Path to the directory for the CSS. |
|----------------|------------------------------------|

**14.15.3.51 printEvents()**

```
void FamilyTree::printEvents (
    std::ostream & os ) [private]
```

Print all events.

## Parameters

|           |                      |
|-----------|----------------------|
| <i>os</i> | Given output stream. |
|-----------|----------------------|

**14.15.3.52 printFiles()**

```
void FamilyTree::printFiles (
    std::ostream & os,
    const std::map< size_t, std::unique_ptr< File > > & container,
    const std::string & label ) [private]
```

Print all files from vector container.

## Parameters

|                  |                                 |
|------------------|---------------------------------|
| <i>os</i>        | Given stream.                   |
| <i>container</i> | Given container of files.       |
| <i>label</i>     | What is the label in JSON file. |

**14.15.3.53 printHtml()**

```
void FamilyTree::printHtml (
    const std::string & dirPath )
```

Print the whole project to an output directory.

## Parameters

|                |                        |
|----------------|------------------------|
| <i>dirPath</i> | Path to the directory. |
|----------------|------------------------|

**14.15.3.54 printHtmlPerson()**

```
void FamilyTree::printHtmlPerson (
    std::ostream & os,
    bool links,
    Person * p ) [private]
```

Print single person to an output in html format.

## Parameters

|              |  |
|--------------|--|
| <i>os</i>    | Which output stream to use.                    |
| <i>links</i> | True if they should be links to other persons. |
| <i>p</i>     | Which person is to be printed.                 |

**14.15.3.55 printMainPersonHtml()**

```
void FamilyTree::printMainPersonHtml (
    const std::string & outputFile )
```

Print main person in HTML format.

**Parameters**

|                   |              |
|-------------------|--------------|
| <i>outputFile</i> | Output file. |
|-------------------|--------------|

**14.15.3.56 printPeople()**

```
void FamilyTree::printPeople (
    std::ostream & os ) [private]
```

Print all the people into a stream in a semi-structured JSON format.

**Parameters**

|           |               |
|-----------|---------------|
| <i>os</i> | Given stream. |
|-----------|---------------|

**14.15.3.57 printRelations()**

```
void FamilyTree::printRelations (
    std::ostream & os ) [private]
```

Print all relations.

**Parameters**

|           |                      |
|-----------|----------------------|
| <i>os</i> | Given output stream. |
|-----------|----------------------|

**14.15.3.58 promoteRelation()**

```
void FamilyTree::promoteRelation (
    size_t id )
```

Promote a relationship to be the persons main. If possible it will be applied to both persons.

**Parameters**

|           |  |
|-----------|--|
| <i>id</i> | Id of the relation that is to be promoted. |
|-----------|--|

#### 14.15.3.59 readJsonEvents()

```
void FamilyTree::readJsonEvents (
    const Json::Value & value ) [private]
```

Read and load all events from its JSON value.

##### Parameters

|              |                                    |
|--------------|------------------------------------|
| <i>value</i> | Loaded value from the source file. |
|--------------|------------------------------------|

#### 14.15.3.60 readJsonFiles()

```
void FamilyTree::readJsonFiles (
    const Json::Value & value,
    std::map< size_t, std::unique_ptr< File > > & container,
    const std::string & label,
    size_t & index ) [private]
```

Read and load meta-data about files from JSON value.

##### Parameters

|                  |  |
|------------------|--|
| <i>value</i>     | Loaded value from the source file.     |
| <i>container</i> | Which vector to use for storing files. |
| <i>label</i>     | Which label was used in the JSON file. |
| <i>index</i>     | Indexing of given container.           |

#### 14.15.3.61 readJsonPeople()

```
void FamilyTree::readJsonPeople (
    const Json::Value & value ) [private]
```

Read and load people from given JSON value.

##### Parameters

|              |                             |
|--------------|-----------------------------|
| <i>value</i> | Loaded value from the file. |
|--------------|-----------------------------|

#### 14.15.3.62 readJsonRelations()

```
void FamilyTree::readJsonRelations (
    const Json::Value & value ) [private]
```

Read and load all relations from its JSON value.

## Parameters

|              |                                    |
|--------------|------------------------------------|
| <i>value</i> | Loaded value from the source file. |
|--------------|------------------------------------|

**14.15.3.63 removeBackup()**

```
void FamilyTree::removeBackup ( )
```

Remove backup file if there is any.

**14.15.3.64 removeEvent()**

```
void FamilyTree::removeEvent (
    size_t id )
```

Remove event from family tree.

## Parameters

|           |                  |
|-----------|------------------|
| <i>id</i> | Id of the event. |
|-----------|------------------|

**14.15.3.65 removeEventTemplate()**

```
void FamilyTree::removeEventTemplate (
    size_t id )
```

Remove event template and all events based on this template.

## Parameters

|           |                     |
|-----------|---------------------|
| <i>id</i> | Id of the template. |
|-----------|---------------------|

**14.15.3.66 removeGeneralOrphanFile()**

```
size_t FamilyTree::removeGeneralOrphanFile (
    std::map< size_t, std::unique_ptr< File > > & container,
    FileType type,
    const std::string & dir ) [private]
```

Remove general orphan file.

## Parameters

|                  |  |
|------------------|--|
| <i>container</i> | Which container holding files is to be used. |
| <i>type</i>      | Which type will be used.                     |
| <i>dir</i>       | Which directory in database to use.          |

**Returns**

Number of remove files.

**14.15.3.67 removeOrphanFiles()**

```
size_t FamilyTree::removeOrphanFiles ( )
```

Remove files that no one uses.

**Returns**

Number of deleted files.

**14.15.3.68 removePerson()**

```
void FamilyTree::removePerson ( )
```

Remove main person from the tree. Also delete all of his appearances.

**14.15.3.69 removeRelation()**

```
void FamilyTree::removeRelation (
    size_t id )
```

Remove relation from the family tree.

**Parameters**

|           |   |
|-----------|---|
| <i>id</i> | Id of the relation which will be removed. |
|-----------|---|

**14.15.3.70 removeRelationTemplate()**

```
void FamilyTree::removeRelationTemplate (
    size_t id )
```

Remove relation template and all relations based on this template.

**Parameters**

|           |                     |
|-----------|---------------------|
| <i>id</i> | Id of the template. |
|-----------|---------------------|

**14.15.3.71 renameFile()**

```
void FamilyTree::renameFile (
    size_t id,
```

```
const std::string & newFilename,  
FileType type )
```

Rename given file.

#### Parameters

|                    |                             |
|--------------------|-----------------------------|
| <i>id</i>          | Id of the file.             |
| <i>newFilename</i> | New file name for the file. |
| <i>type</i>        | What type is the file.      |

#### 14.15.3.72 restoreBackup()

```
bool FamilyTree::restoreBackup (   
    const std::string & backupFile,  
    std::string & errorMessage )
```

Restore backup file from the database. Also load the project again.

#### Parameters

|                     |                                  |
|---------------------|----------------------------------|
| <i>backupFile</i>   | Which is the backup file.        |
| <i>errorMessage</i> | If error occurred show it there. |

#### Returns

If the parsing of restored database was successful or not.

#### 14.15.3.73 setMainPerson() [1/2]

```
void FamilyTree::setMainPerson (   
    Person * p )
```

Set the main person.

#### Parameters

|          |                  |
|----------|------------------|
| <i>p</i> | New main person. |
|----------|------------------|

#### 14.15.3.74 setMainPerson() [2/2]

```
void FamilyTree::setMainPerson (   
    size_t id )
```

Set the main person.

**Parameters**

|           |                   |
|-----------|-------------------|
| <i>id</i> | Id of the person. |
|-----------|-------------------|

**14.15.3.75 setRelation()**

```
void FamilyTree::setRelation (
    size_t relId,
    size_t pers1Id,
    size_t pers2Id,
    size_t templId )
```

Set the data about one relation.

**Parameters**

|                             |                                  |
|-----------------------------|----------------------------------|
| <i>relId</i>                | Id of the relation to be edited. |
| <i>pers1</i> ↔<br><i>Id</i> | Id of the first person.          |
| <i>pers2</i> ↔<br><i>Id</i> | Id of the second person.         |
| <i>templ</i> ↔<br><i>Id</i> | Id of the used template.         |

**14.15.3.76 setUnsaved()**

```
void FamilyTree::setUnsaved ( )
```

Set unsaved state.

**14.15.3.77 showHelpOnStartup()**

```
bool FamilyTree::showHelpOnStartup ( )
```

If help dialog should be shown.

**Returns**

True if it should be shown.

**14.15.3.78 storeDatabase()**

```
bool FamilyTree::storeDatabase ( )
```

Store the whole database to its files.

**Returns**

If the storing was successful.



**14.15.3.79 storeFileDatabase()**

```
void FamilyTree::storeFileDatabase (
    FileType type ) [private]
```

Store file database when putting file to database. Also remove all orphan files.

## Parameters

|             |                               |
|-------------|-------------------------------|
| <i>type</i> | What type of files are saved. |
|-------------|-------------------------------|

**14.15.3.80 templatesBasedOnEventTemplate()**

```
size_t FamilyTree::templatesBasedOnEventTemplate (
    size_t templateId )
```

Get the number of events based on this event template.

## Parameters

|                         |                           |
|-------------------------|---------------------------|
| <i>template↔<br/>Id</i> | Id of the event template. |
|-------------------------|---------------------------|

## Returns

Number of events using this event template.

**14.15.3.81 templatesBasedOnRelationTemplate()**

```
size_t FamilyTree::templatesBasedOnRelationTemplate (
    size_t templateId )
```

Get the number of relations based on this relation template.

## Parameters

|                         |                              |
|-------------------------|------------------------------|
| <i>template↔<br/>Id</i> | Id of the relation template. |
|-------------------------|------------------------------|

## Returns

Number of relations using this relation template.

**14.15.3.82 updateEventsWithTemplate()**

```
void FamilyTree::updateEventsWithTemplate (
    size_t templ )
```

Update events based on the changed template.

## Parameters

|              |                             |
|--------------|-----------------------------|
| <i>templ</i> | Id of the changed template. |
|--------------|-----------------------------|

### 14.15.3.83 writeConfig()

```
bool FamilyTree::writeConfig (
    const std::string & newPath )
```

Write to the configuration file and before add new one if it is not present.

#### Parameters

|                |                          |
|----------------|--------------------------|
| <i>newPath</i> | Path to the new project. |
|----------------|--------------------------|

#### Returns

True if it was added. False if it was already present.

## 14.15.4 Member Data Documentation

### 14.15.4.1 allEvents\_

```
std::map<size_t, std::unique_ptr<Event> > FamilyTree::allEvents_ [private]
```

All the events in the tree.

### 14.15.4.2 allFiles\_

```
std::map<size_t, std::unique_ptr<File> > FamilyTree::allFiles_ [private]
```

All files in app.

### 14.15.4.3 allMedia\_

```
std::map<size_t, std::unique_ptr<File> > FamilyTree::allMedia_ [private]
```

All media in app.

### 14.15.4.4 allNotes\_

```
std::map<size_t, std::unique_ptr<File> > FamilyTree::allNotes_ [private]
```

All notes in app.

### 14.15.4.5 allPersons\_

```
std::map<size_t, std::unique_ptr<Person> > FamilyTree::allPersons_ [private]
```

Map of all persons in family tree.

#### 14.15.4.6 allRelations\_

```
std::map<size_t, std::unique_ptr<Relation> > FamilyTree::allRelations_ [private]
```

All relations in the tree.

#### 14.15.4.7 event\_index\_

```
size_t FamilyTree::event_index_ [private]
```

Last free index for event. Always start from 1.

#### 14.15.4.8 file\_index\_

```
size_t FamilyTree::file_index_ [private]
```

Last free id for file. Always start from 1.

#### 14.15.4.9 mainPerson\_

```
Person* FamilyTree::mainPerson_ [private]
```

Main person showing as the centre of the tree.

#### 14.15.4.10 media\_index\_

```
size_t FamilyTree::media_index_ [private]
```

Last free id for media. Always start from 1.

#### 14.15.4.11 note\_index\_

```
size_t FamilyTree::note_index_ [private]
```

Last free id for note. Always start from 1.

#### 14.15.4.12 openHelp

```
bool FamilyTree::openHelp [private]
```

If help window should be shown.

#### 14.15.4.13 parser\_

```
Parser FamilyTree::parser_ [private]
```

[Parser](#) for working with locally stored database.

#### 14.15.4.14 person\_index\_

```
size_t FamilyTree::person_index_ [private]
```

Last free id for person. Always start from 1.

#### 14.15.4.15 projectPaths

```
std::vector<std::string> FamilyTree::projectPaths [private]
```

Paths to existing projects.

#### 14.15.4.16 relation\_index\_

```
size_t FamilyTree::relation_index_ [private]
```

Last free index for relation.

#### 14.15.4.17 savedChanges\_

```
bool FamilyTree::savedChanges_ [private]
```

If all changes were saved or not.

#### 14.15.4.18 settings\_

```
Settings FamilyTree::settings_ [private]
```

[Settings](#) of the app.

The documentation for this class was generated from the following files:

- [core/family\\_tree.h](#)
- [core/family\\_tree.cpp](#)

## 14.16 File Class Reference

Meta-data about one file (media or note). Its real path is [id]-[filename].

```
#include <family_tree_items.h>
```

## Public Member Functions

- [File](#) ()  
*Default constructor.*
- const std::string & [getFilename](#) () const
- size\_t [getId](#) () const
- std::string [getRealName](#) () const
- void [readJson](#) (const Json::Value &value)
- void [setFilename](#) (const std::string &fileName)
- void [setId](#) (size\_t id)

## Private Attributes

- std::string [filename\\_](#)  
*File name.*
- size\_t [id\\_](#)  
*Id of the file.*

### 14.16.1 Detailed Description

Meta-data about one file (media or note). Its real path is [id]-[filename].

### 14.16.2 Constructor & Destructor Documentation

#### 14.16.2.1 File()

```
File::File ( )
```

Default constructor.

### 14.16.3 Member Function Documentation

#### 14.16.3.1 getFilename()

```
const std::string & File::getFilename ( ) const
```

Get the file name.

#### Returns

Constant reference to the file name.

### 14.16.3.2 getId()

```
size_t File::getId ( ) const
```

Get the id of the file.

#### Returns

Id of the file.

### 14.16.3.3 getRealName()

```
std::string File::getRealName ( ) const
```

Get the real name of the file. Which is [id]-[filename]

#### Returns

The real name of the file.

### 14.16.3.4 readJson()

```
void File::readJson (
    const Json::Value & value )
```

Read the file meta-data from JSON value.

#### Parameters

|              |                             |
|--------------|-----------------------------|
| <i>value</i> | The JSON value of the file. |
|--------------|-----------------------------|

### 14.16.3.5 setFilename()

```
void File::setFilename (
    const std::string & fileName )
```

Set the file name.

#### Parameters

|                 |                         |
|-----------------|-------------------------|
| <i>fileName</i> | Given name of the file. |
|-----------------|-------------------------|

### 14.16.3.6 setId()

```
void File::setId (
    size_t id )
```

Set the id of the file.



## Parameters

|           |           |
|-----------|-----------|
| <i>id</i> | Given id. |
|-----------|-----------|

## 14.16.4 Member Data Documentation

### 14.16.4.1 filename\_

```
std::string File::filename_ [private]
```

File name.

### 14.16.4.2 id\_

```
size_t File::id_ [private]
```

Id of the file.

The documentation for this class was generated from the following files:

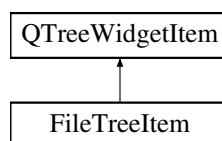
- [core/family\\_tree\\_items.h](#)
- [core/family\\_tree\\_items.cpp](#)

## 14.17 FileTreeItem Class Reference

TreeWidget Item for general file.

```
#include <graphics_items.h>
```

Inheritance diagram for FileTreeItem:



### Public Member Functions

- [FileTreeItem](#) ([VirtualDrive](#) \*drive, [FamilyTree](#) \*FT, const [QIcon](#) &icon, [FileType](#) type)
- [FileTreeItem](#) (size\_t id, [FamilyTree](#) \*FT, const [QIcon](#) &icon, [FileType](#) type)
- void [addFile](#) (size\_t id, const [QIcon](#) &icon)
- void [addSubDrive](#) (const [QString](#) &folder)
- [VirtualDrive](#) \* [getDrive](#) ()
- size\_t [getId](#) ()
- bool [isDirectory](#) ()
- bool [removeFolder](#) ()
- bool [rename](#) (const [QString](#) &name)

## Private Member Functions

- void [removeFile](#) (size\_t id)

## Private Attributes

- [VirtualDrive](#) \* [drive\\_](#)  
*Drive presented in this folder.*
- size\_t [fileId\\_](#)  
*Id of the shown file.*
- [FamilyTree](#) \* [FT\\_](#)  
*Pointer to the family tree.*
- bool [isDir\\_](#)  
*If the file is directory or not.*
- [FileType](#) [type\\_](#)  
*Type of the file.*

## 14.17.1 Detailed Description

TreeWidget Item for general file.

## 14.17.2 Constructor & Destructor Documentation

### 14.17.2.1 FileTreeItem() [1/2]

```
FileTreeItem::FileTreeItem (
    VirtualDrive * drive,
    FamilyTree * FT,
    const QIcon & icon,
    FileType type )
```

Construct this drive and all sub-drives.

#### Parameters

|              |  |
|--------------|--|
| <i>drive</i> | Which drive is presented in this item. |
| <i>FT</i>    | Pointer to the family tree.            |
| <i>icon</i>  | Which icon to show beside the text.    |
| <i>type</i>  | Type of the file to show.              |

### 14.17.2.2 FileTreeItem() [2/2]

```
FileTreeItem::FileTreeItem (
    size_t id,
    FamilyTree * FT,
    const QIcon & icon,
    FileType type )
```

Constructor for a new file.

**Parameters**

|             |                                    |
|-------------|------------------------------------|
| <i>id</i>   | Id of the file.                    |
| <i>FT</i>   | Pointer to the family tree.        |
| <i>icon</i> | Which icon is shown besides files. |
| <i>type</i> | What type of files are shown.      |

## 14.17.3 Member Function Documentation

### 14.17.3.1 addFile()

```
void FileTreeItem::addFile (
    size_t id,
    const QIcon & icon )
```

Add file to this folder.

**Parameters**

|             |                                      |
|-------------|--------------------------------------|
| <i>id</i>   | Id of the file.                      |
| <i>icon</i> | Which icon to show besides the text. |

### 14.17.3.2 addSubDrive()

```
void FileTreeItem::addSubDrive (
    const QString & folder )
```

Add sub drive to this drive.

**Parameters**

|               |                             |
|---------------|-----------------------------|
| <i>folder</i> | The name of the sub-folder. |
|---------------|-----------------------------|

### 14.17.3.3 getDrive()

```
VirtualDrive * FileTreeItem::getDrive ( )
```

Get the pointer to the drive.

**Returns**

Pointer to the drive.

#### 14.17.3.4 getId()

```
size_t FileTreeItem::getId ( )
```

Get the id of the file.

Returns

File id.

#### 14.17.3.5 isDirectory()

```
bool FileTreeItem::isDirectory ( )
```

If te file is directory or not.

Returns

True if it is directory, false otherwise.

#### 14.17.3.6 removeFile()

```
void FileTreeItem::removeFile (
    size_t id ) [private]
```

Remove file in this drive.

Parameters

|           |                 |
|-----------|-----------------|
| <i>id</i> | Id of the file. |
|-----------|-----------------|

#### 14.17.3.7 removeFolder()

```
bool FileTreeItem::removeFolder ( )
```

Remove recursively all directories and files.

Returns

True if it is the root directory, false otherwise.

#### 14.17.3.8 rename()

```
bool FileTreeItem::rename (
    const QString & name )
```

When this item is selected for renaming.

#### Parameters

|             |                       |
|-------------|-----------------------|
| <i>name</i> | Name for this folder. |
|-------------|-----------------------|

#### Returns

True if this item was file, false otherwise.

### 14.17.4 Member Data Documentation

#### 14.17.4.1 drive\_

`VirtualDrive* FileTreeItem::drive_ [private]`

Drive presented in this folder.

#### 14.17.4.2 fileId\_

`size_t FileTreeItem::fileId_ [private]`

Id of the shown file.

#### 14.17.4.3 FT\_

`FamilyTree* FileTreeItem::FT_ [private]`

Pointer to the family tree.

#### 14.17.4.4 isDir\_

`bool FileTreeItem::isDir_ [private]`

If the file is directory or not.

#### 14.17.4.5 type\_

`FileType FileTreeItem::type_ [private]`

Type of the file.

The documentation for this class was generated from the following files:

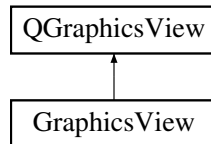
- [graphics/graphics\\_items.h](#)
- [graphics/graphics\\_items.cpp](#)

## 14.18 QGraphicsView Class Reference

Class for creating QGraphicsView with using wheel for zoom in and out. And grabing by mouse moves the view.

```
#include <graphics_items.h>
```

Inheritance diagram for QGraphicsView:



### Public Member Functions

- [GraphicsView](#) (QGraphicsScene \*scene, QWidget \*parent=nullptr)

### Protected Member Functions

- void [wheelEvent](#) (QWheelEvent \*event) override

### 14.18.1 Detailed Description

Class for creating QGraphicsView with using wheel for zoom in and out. And grabing by mouse moves the view.

### 14.18.2 Constructor & Destructor Documentation

#### 14.18.2.1 GraphicsView()

```
GraphicsView::GraphicsView (
    QGraphicsScene * scene,
    QWidget * parent = nullptr ) [explicit]
```

Default constructor.

#### Parameters

|               |  |
|---------------|--|
| <i>scene</i>  | Which scene it is using.               |
| <i>parent</i> | Which widget will be showing the view. |

### 14.18.3 Member Function Documentation

#### 14.18.3.1 wheelEvent()

```
void GraphicsView::wheelEvent (
    QWheelEvent * event ) [override], [protected]
```

Override the wheel event for scrolling.

#### Parameters

|              |                          |
|--------------|--------------------------|
| <i>event</i> | When scrolling occurred. |
|--------------|--------------------------|

The documentation for this class was generated from the following files:

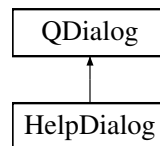
- [graphics/graphics\\_items.h](#)
- [graphics/graphics\\_items.cpp](#)

## 14.19 HelpDialog Class Reference

Dialog with text edit to show help or other texts.

```
#include <dialogs.h>
```

Inheritance diagram for HelpDialog:



#### Public Slots

- void [openHelp](#) (bool help)

#### Public Member Functions

- [HelpDialog](#) ([FamilyTree](#) \*FT, QWidget \*parent=nullptr)
- [~HelpDialog](#) ()

*Default destructor.*

#### Private Attributes

- [FamilyTree](#) \* FT\_  
*Pointer to the family tree.*
- Ui::HelpDialog \* ui  
*User interface of the Qt framework.*

### 14.19.1 Detailed Description

Dialog with text edit to show help or other texts.

### 14.19.2 Constructor & Destructor Documentation

#### 14.19.2.1 HelpDialog()

```
HelpDialog::HelpDialog (
    FamilyTree * FT,
    QWidget * parent = nullptr ) [explicit]
```

Default constructor.



## Parameters

|               |                             |
|---------------|-----------------------------|
| <i>FT</i>     | Pointer to the family tree. |
| <i>parent</i> | The Qt Widget parent.       |

**14.19.2.2 ~HelpDialog()**

```
HelpDialog::~HelpDialog ( )
```

Default destructor.

**14.19.3 Member Function Documentation****14.19.3.1 openHelp**

```
void HelpDialog::openHelp (
    bool help ) [slot]
```

If help dialog should be opened at start-up.

## Parameters

|             |  |
|-------------|--|
| <i>help</i> | Forwarding of check box in UI. True for not showing. |
|-------------|--|

**14.19.4 Member Data Documentation****14.19.4.1 FT\_**

```
FamilyTree* HelpDialog::FT_ [private]
```

Pointer to the family tree.

**14.19.4.2 ui**

```
Ui::HelpDialog* HelpDialog::ui [private]
```

User interface of the Qt framework.

The documentation for this class was generated from the following files:

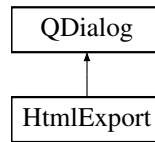
- [graphics/dialogs.h](#)
- [graphics/dialogs.cpp](#)

## 14.20 HtmlExport Class Reference

Dialog for exporting HTML for one person and all persons.

```
#include <dialogs.h>
```

Inheritance diagram for HtmlExport:



### Public Slots

- void [allAboveGenerations](#) (bool all)
- void [allBelowGenerations](#) (bool all)
- void [exportAllHtml](#) ()  
*Export all persons into html.*
- void [exportMainHtml](#) ()  
*Export only main person to the html.*

### Public Member Functions

- [HtmlExport](#) ([FamilyTree](#) \*FT, bool all, int up, int down, QWidget \*parent=nullptr)
- [~HtmlExport](#) ()  
*Default destructor.*

### Private Attributes

- [FamilyTree](#) \* FT\_  
*Pointer to the family tree.*
- Ui::HtmlExport \* ui  
*Used Qt user interface.*

### 14.20.1 Detailed Description

Dialog for exporting HTML for one person and all persons.

### 14.20.2 Constructor & Destructor Documentation

#### 14.20.2.1 HtmlExport()

```

HtmlExport::HtmlExport (
    FamilyTree * FT,
    bool all,
    int up,
    int down,
    QWidget * parent = nullptr )

```

Default constructor.

## Parameters

|               |  |
|---------------|--|
| <i>FT</i>     | Pointer to the used Family tree.                                       |
| <i>all</i>    | If the HTML export should include all persons or just the main person. |
| <i>up</i>     | How many generations upwards are being exported as of now.             |
| <i>down</i>   | How many generations downwards are being exported as of now.           |
| <i>parent</i> | The Qt Widget parent.  |

**14.20.2.2 ~HtmlExport()**

```
HtmlExport::~HtmlExport ( )
```

Default destructor.

**14.20.3 Member Function Documentation****14.20.3.1 allAboveGenerations**

```
void HtmlExport::allAboveGenerations (
    bool all ) [slot]
```

If all generations upward option was chosen or not.

## Parameters

|            |                         |
|------------|-------------------------|
| <i>all</i> | True if it was checked. |
|------------|-------------------------|

**14.20.3.2 allBelowGenerations**

```
void HtmlExport::allBelowGenerations (
    bool all ) [slot]
```

If all generations downward option was chosen or not.

## Parameters

|            |                         |
|------------|-------------------------|
| <i>all</i> | True if it was checked. |
|------------|-------------------------|

**14.20.3.3 exportAllHtml**

```
void HtmlExport::exportAllHtml ( ) [slot]
```

Export all persons into html.

#### 14.20.3.4 exportMainHtml

```
void HtmlExport::exportMainHtml ( ) [slot]
```

Export only main person to the html.

### 14.20.4 Member Data Documentation

#### 14.20.4.1 FT\_

```
FamilyTree* HtmlExport::FT_ [private]
```

Pointer to the family tree.

#### 14.20.4.2 ui

```
Ui::HtmlExport* HtmlExport::ui [private]
```

Used Qt user interface.

The documentation for this class was generated from the following files:

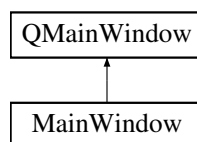
- [graphics/dialogs.h](#)
- [graphics/dialogs.cpp](#)

## 14.21 MainWindow Class Reference

Class representing main window using Qt library.

```
#include <mainwindow.h>
```

Inheritance diagram for MainWindow:



## Public Slots

- void [addNewEvent](#) ()  
*Show dialog for adding event.*
- void [addNewFile](#) ()  
*Show dialog for adding file.*
- void [addNewFileFolder](#) ()  
*Add new folder to the file tree.*
- void [addNewMedia](#) ()  
*Show dialog for adding media.*
- void [addNewMediaFolder](#) ()  
*Add new folder to the media tree.*
- void [addNewNote](#) ()  
*Show dialog for adding note.*
- void [addNewNoteFolder](#) ()  
*Add new folder to the note tree.*
- void [addNewPerson](#) ()  
*Add new person to the family tree.*
- void [addNewRelation](#) ()  
*Add new relation to the person by opening dialog window.*
- void [allAboveGenerations](#) (bool selected)
- void [allBelowGenerations](#) (bool selected)
- void [changeGenerationsDown](#) (int value)
- void [changeGenerationsUp](#) (int value)
- void [checkStoreRefreshPersonsInfo](#) ()  
*If any value in persons info tab was changed, check its value, store it and refresh ui elements.*
- void [clearProjectFilter](#) ()  
*Clear the project filter.*
- void [clearProjectPaths](#) ()  
*Clear project paths.*
- void [disableFileFolderPrompt](#) ()  
*Hide new folder edit for files and cancel button.*
- void [disableMediaFolderPrompt](#) ()  
*Hide new folder edit for media and cancel button.*
- void [disableNoteFolderPrompt](#) ()  
*Hide new folder edit for notes and cancel button.*
- void [editEvent](#) (QTreeWidgetItem \*item)
- void [editRelation](#) (QTreeWidgetItem \*item)
- void [editTag](#) (QTreeWidgetItem \*item=nullptr)
- void [enableFileFolderPrompt](#) ()  
*Show new folder edit for files and cancel button.*
- void [enableMediaFolderPrompt](#) ()  
*Show new folder edit for media and cancel button.*
- void [enableNoteFolderPrompt](#) ()  
*Show new folder edit for notes and cancel button.*
- void [exportCloseFamilyDiagramDialog](#) ()  
*Open dialog for exporting close family diagram.*
- void [exportHtml](#) ()  
*Export whole HTML project to an output directory by opening a dialog.*
- void [exportMainHtml](#) ()  
*Export only main person html by opening a dialog.*

- void [exportProject](#) ()  
*Open dialog for exporting project.*
- void [exportTemplates](#) ()  
*Open dialog for exporting templates.*
- void [exportTreeDiagramDialog](#) ()  
*Open dialog for exporting family tree diagram.*
- void [filterProjectItems](#) ()  
*Filter only matching items in project.*
- void [findPerson](#) ()  
*Focus find in project view.*
- void [fitTreeInView](#) ()  
*Fit the tree view inside.*
- void [importProject](#) ()  
*Choose file to import from other project.*
- void [importRootDir](#) ()  
*Choose root directory of other project to import from.*
- void [importTemplates](#) ()  
*Choose file to import templates.*
- void [interactWithProjectItem](#) (QTreeWidgetItem \*item)
- void [openBirthDateDialog](#) ()  
*Open dialog for custom Birth [Date](#).*
- void [openDeathDateDialog](#) ()  
*Open dialog for custom Death [Date](#).*
- void [openEventTemplatesDialog](#) ()  
*Open event templates dialog.*
- void [openExistingProject](#) ()  
*Open existing root directory of a project.*
- void [openGraphicSettings](#) ()  
*Open dialog for graphic settings.*
- void [openNewFamilyTreeProject](#) ()  
*Open new empty project.*
- void [openRelationTemplatesDialog](#) ()  
*Open relation templates dialog.*
- void [openSelectedFile](#) (QTreeWidgetItem \*item)
- void [openSelectedMedia](#) (QTreeWidgetItem \*item)
- void [openSelectedNote](#) (QTreeWidgetItem \*item)
- void [promoteRelation](#) ()  
*Promote current relation.*
- void [refreshDates](#) ()  
*Refresh all dates in persons info tab.*
- void [refreshEventTab](#) ()  
*Refresh list with events.*
- void [refreshGraphics](#) ()  
*Refresh all graphics.*
- void [refreshProjectView](#) ()  
*Refresh the whole project that is shown.*
- void [refreshRelationTab](#) ()  
*Refresh relations lists.*
- void [refreshTabsWithFiles](#) ()  
*Refresh all files tabs.*
- void [refreshTagsList](#) ()

- Refresh shown tags.*
- void [removeEvent](#) ()  
*Remove event from family tree.*
- void [removeFileFolder](#) ()  
*Remove file folder.*
- void [removeMediaFolder](#) ()  
*Remove media folder.*
- void [removeNoteFolder](#) ()  
*Remove note folder.*
- void [removePerson](#) ()  
*Remove main person.*
- void [removeRelation](#) ()  
*Remove current relation.*
- void [removeTag](#) ()  
*Delete current tag selected on the person.*
- void [renameFile](#) ()  
*Rename file or folder with files.*
- void [renameMedia](#) ()  
*Rename media or folder with media.*
- void [renameNote](#) ()  
*Rename note or folder with notes.*
- bool [saveDatabase](#) ()
- void [saveGraphicsSetting](#) (QColor probandColor, QColor promotedColor, QColor standardColor, QColor highlightedColor, int sizeX, int sizeY, QFont font, QColor lineColor, QColor textColor, int lineWidth, int rounding)
- void [selectOlderGenerationOnly](#) ()  
*De-select all other relation boxes other than older.*
- void [selectSameGenerationOnly](#) ()  
*De-select all other relation boxes other than same.*
- void [selectYoungerGenerationOnly](#) ()  
*De-select all other relation boxes other than younger.*
- void [setEnabledDeathAtributes](#) (bool disabled)
- void [setEventTabVisible](#) ()  
*Hide or show the event tab.*
- void [setFileTabVisible](#) ()  
*Hide or show the file tab.*
- void [setGeneralTabVisible](#) ()  
*Hide or show the general tab.*
- void [setMainPerson](#) (size\_t personId)
- void [setMediaTabVisible](#) ()  
*Hide or show the media tab.*
- void [setNoteTabVisible](#) ()  
*Hide or show the note tab.*
- void [setRelationTabVisible](#) ()  
*Hide or show the relation tab.*
- void [setTabEast](#) ()  
*Set the setting of the tab to east.*
- void [setTabNorth](#) ()  
*Set the setting of the tab to north.*
- void [setTabSouth](#) ()  
*Set the setting of the tab to south.*

- void [setTabWest](#) ()  
*Set the setting of the tab to west.*
- void [showHelp](#) ()  
*Show generic help.*
- void [showRenameFile](#) ()  
*Show prompt for renaming file.*
- void [showRenameMedia](#) ()  
*Show prompt for renaming media.*
- void [showRenameNote](#) ()  
*Show prompt for renaming note.*
- void [splitterChanged](#) ()  
*When the splitter changed.*
- void [switchNewTreeProject](#) ()  
*Save tree and create new empty tree to show.*
- void [zoomIn](#) ()  
*Zoom in on the family tree view.*
- void [zoomOut](#) ()  
*Zoom out of the family tree view.*

### Public Member Functions

- [MainWindow](#) (QWidget \*parent=nullptr)
- [~MainWindow](#) ()  
*Default destructor, which deletes all dynamic attributes.*
- bool [exportCloseFamilyDiagram](#) (double scale, const std::string &filename=[EMPTY\\_STRING](#))
- bool [exportTreeDiagram](#) (double scale, const std::string &filename=[EMPTY\\_STRING](#))
- std::pair< int, int > [setNumberOfGenerations](#) (int up, int down)

### Static Public Member Functions

- static void [openExternalFile](#) (const std::string &fileURL)

### Protected Member Functions

- void [closeEvent](#) (QCloseEvent \*event)

### Private Member Functions

- void [addGeneralFile](#) ([FileType](#) type, const std::vector< std::string > &extensions, [FileTreeItem](#) \*item)
- void [addGeneralFolder](#) ([FileType](#) type)
- void [clearUi](#) ()  
*Clear all UI elements.*
- void [connectAllSlots](#) ()  
*Connect all slots to their signals.*
- void [editEvent](#) (size\_t id)
- void [editRelation](#) (size\_t id)
- void [enableGeneralFolderPrompt](#) (bool visible, [FileType](#) type)
- void [enableGeneralFolderRenamePrompt](#) ([FileType](#) type)
- bool [exportGeneralDiagram](#) (const QString &filename, QGraphicsScene \*gscene, double scale=1)



- void [drawCloseContainer](#) (const std::vector< [Relation](#) \* > &container, qreal sizeX, qreal sizeY, qreal topLeftX, qreal topLeftY, qreal lineX1, qreal lineY1, qreal lineX2, qreal lineY2, size\_t partner, size\_t mother, size\_t father, int gap, bool movex, bool movey)
- void [drawCloseFamily](#) ()
- void [drawFamilyTree](#) ()
- void [drawGeneralGenerations](#) ([Person](#) \*p, qreal width, qreal midx, qreal midy, int horizontalGap, int verticalGap, size\_t barrier, const std::vector< size\_t > &container, bool up, std::function< std::pair< size\_t, size\_t >(< [Person](#) \*, size\_t, size\_t)> numberFunc, std::function< void([Person](#) \*, qreal, qreal, qreal, int, int, size\_t)> drawFunc)
- void [drawOlderGenerations](#) ([Person](#) \*p, qreal width, qreal midx, qreal midy, int horizontalGap, int verticalGap, size\_t barrier)
- void [drawPartner](#) (qreal canvasMidX, qreal canvasMidY, int horizontalGap)
- void [drawYoungerGenerations](#) ([Person](#) \*p, qreal width, qreal midx, qreal midy, int horizontalGap, int verticalGap, size\_t barrier)
- std::pair< size\_t, size\_t > [generalGenerationNumber](#) ([Person](#) \*p, size\_t soFar, size\_t barrier, const std::vector< size\_t > &container, bool up, std::function< std::pair< size\_t, size\_t >(< [Person](#) \*, size\_t, size\_t)> func)
- void [loadSettings](#) ()
  - Load the setting from configuration.*
- std::pair< size\_t, size\_t > [olderGenerationNumber](#) ([Person](#) \*p, int soFar, size\_t barrier)
- void [openCustomDateDialog](#) ([WrappedDate](#) \*date)
- void [initializeProjectSubView](#) ([ProjectItem](#) \*parent, [Person](#) \*person)
- void [initializeProjectView](#) ()
  - Initialize project view when loading new family tree.*
- void [openGeneralFile](#) (QTreeWidgetItem \*item, [FileType](#) type)
- void [openGeneralProject](#) (const std::string &dirPath)
- void [refreshPersonsInfoTab](#) ()
  - Refresh basic information about main person.*
- void [refreshProjectEventsView](#) ([ProjectItem](#) \*item, [Person](#) \*person)
- void [refreshProjectGeneralFilesView](#) (const QIcon &containerIcon, const QString &containerName, const QIcon &fileIcon, [FileType](#) type, [ProjectItem](#) \*item, [Person](#) \*person)
- void [refreshProjectFilesView](#) ([ProjectItem](#) \*item, [Person](#) \*person)
- void [refreshProjectMediaView](#) ([ProjectItem](#) \*item, [Person](#) \*person)
- void [refreshProjectNoteView](#) ([ProjectItem](#) \*item, [Person](#) \*person)
- void [refreshProjectRelationsView](#) ([ProjectItem](#) \*item, [Person](#) \*person)
- void [refreshUi](#) ()
  - Refresh the whole UI (tabs, project and graphics).*
- void [removeGeneralFolder](#) ([FileType](#) type)
- void [renameGeneralFileItem](#) ([FileType](#) type)
- void [savePersonsInfo](#) ()
  - Save all basic information about main person.*
- void [setUiElements](#) ()
  - Set all additional static widgets of the user interface.*
- void [showGeneralFiles](#) (QWidget \*widget, [VirtualDrive](#) \*root, [FileType](#) type)
- std::pair< size\_t, size\_t > [youngerGenerationNumber](#) ([Person](#) \*p, size\_t soFar, size\_t barrier)

## Private Attributes

- int [borderRadius\\_](#)
  - Radius of the boxes.*
- QFont [font\\_](#)
  - Used font for view.*
- [FamilyTree](#) FT

- Family tree.*
- int [genSizeDown\\_](#)  
*How many generations to show in tree view down.*
- int [genSizeUp\\_](#)  
*How many generations to show in tree view up.*
- QColor [highlightedColor\\_](#)  
*Color for highlighted boxes.*
- QPen [linePen\\_](#)  
*Pen used for drawing lines.*
- QColor [probandColor\\_](#)  
*Color of proband.*
- QGraphicsScene \* [scene](#)  
*Scene shown on the graphical view.*
- int [sizeX\\_](#)  
*X size of the person box.*
- int [sizeY\\_](#)  
*Y size of the person box.*
- QColor [promotedColor\\_](#)  
*Color for promoted father, mother and partner.*
- QColor [standardColor\\_](#)  
*Color for other relations.*
- QPen [textPen\\_](#)  
*Pen used for texts.*
- QGraphicsScene \* [treeScene](#)  
*Tree view scene;.*
- [GraphicsView](#) \* [treeView](#)  
*Tree view graphics view.*
- Ui::MainWindow \* [ui](#)  
*Qt UI framework.*

### 14.21.1 Detailed Description

Class representing main window using Qt library.

### 14.21.2 Constructor & Destructor Documentation

#### 14.21.2.1 MainWindow()

```
MainWindow::MainWindow (
    QWidget * parent = nullptr ) [explicit]
```

Default constructor.

Parameters

|               |            |
|---------------|------------|
| <i>parent</i> | Qt parent. |
|---------------|------------|

### 14.21.2.2 ~MainWindow()

```
MainWindow::~MainWindow ( )
```

Default destructor, which deletes all dynamic attributes.

## 14.21.3 Member Function Documentation

### 14.21.3.1 addGeneralFile()

```
void MainWindow::addGeneralFile (
    FileType type,
    const std::vector< std::string > & extensions,
    FileTreeItem * item ) [private]
```

Open add file dialog for a person. This is as general as it gets.

#### Parameters

|                   |   |
|-------------------|---|
| <i>type</i>       | Type of the file, that will be added.                   |
| <i>extensions</i> | Which extensions will be shown for creating empty file. |
| <i>item</i>       | Which item is selected to put file.                     |

### 14.21.3.2 addGeneralFolder()

```
void MainWindow::addGeneralFolder (
    FileType type ) [private]
```

Add folder to its type tree widget.

#### Parameters

|             |                            |
|-------------|----------------------------|
| <i>type</i> | What widget it should use. |
|-------------|----------------------------|

### 14.21.3.3 addNewEvent

```
void MainWindow::addNewEvent ( ) [slot]
```

Show dialog for adding event.

### 14.21.3.4 addNewFile

```
void MainWindow::addNewFile ( ) [slot]
```

Show dialog for adding file.

#### 14.21.3.5 addNewFileFolder

```
void MainWindow::addNewFileFolder ( ) [slot]
```

Add new folder to the file tree.

#### 14.21.3.6 addNewMedia

```
void MainWindow::addNewMedia ( ) [slot]
```

Show dialog for adding media.

#### 14.21.3.7 addNewMediaFolder

```
void MainWindow::addNewMediaFolder ( ) [slot]
```

Add new folder to the media tree.

#### 14.21.3.8 addNewNote

```
void MainWindow::addNewNote ( ) [slot]
```

Show dialog for adding note.

#### 14.21.3.9 addNewNoteFolder

```
void MainWindow::addNewNoteFolder ( ) [slot]
```

Add new folder to the note tree.

#### 14.21.3.10 addNewPerson

```
void MainWindow::addNewPerson ( ) [slot]
```

Add new person to the family tree.

#### 14.21.3.11 addNewRelation

```
void MainWindow::addNewRelation ( ) [slot]
```

Add new relation to the person by opening dialog window.

#### 14.21.3.12 allAboveGenerations

```
void MainWindow::allAboveGenerations (
    bool selected ) [slot]
```

Set to show all generations above.

## Parameters

|                 |                                     |
|-----------------|-------------------------------------|
| <i>selected</i> | If all generations should be shown. |
|-----------------|-------------------------------------|

**14.21.3.13 allBelowGenerations**

```
void MainWindow::allBelowGenerations (
    bool selected ) [slot]
```

Set to show all generations below.

## Parameters

|                 |                                     |
|-----------------|-------------------------------------|
| <i>selected</i> | If all generations should be shown. |
|-----------------|-------------------------------------|

**14.21.3.14 changeGenerationsDown**

```
void MainWindow::changeGenerationsDown (
    int value ) [slot]
```

Change the number of shown relations below.

## Parameters

|              |                               |
|--------------|-------------------------------|
| <i>value</i> | How many generations to show. |
|--------------|-------------------------------|

**14.21.3.15 changeGenerationsUp**

```
void MainWindow::changeGenerationsUp (
    int value ) [slot]
```

Change the number of shown relations above.

## Parameters

|              |                               |
|--------------|-------------------------------|
| <i>value</i> | How many generations to show. |
|--------------|-------------------------------|

**14.21.3.16 checkStoreRefreshPersonsInfo**

```
void MainWindow::checkStoreRefreshPersonsInfo ( ) [slot]
```

If any value in persons info tab was changed, check its value, store it and refresh ui elements.

#### 14.21.3.17 clearProjectFilter

```
void MainWindow::clearProjectFilter ( ) [slot]
```

Clear the project filter.

#### 14.21.3.18 clearProjectPaths

```
void MainWindow::clearProjectPaths ( ) [slot]
```

Clear project paths.

#### 14.21.3.19 clearUi()

```
void MainWindow::clearUi ( ) [private]
```

Clear all UI elements.

#### 14.21.3.20 closeEvent()

```
void MainWindow::closeEvent (
    QCloseEvent * event ) [protected]
```

When closing the window check if the project is saved or not.

##### Parameters

|              |               |
|--------------|---------------|
| <i>event</i> | Called event. |
|--------------|---------------|

#### 14.21.3.21 connectAllSlots()

```
void MainWindow::connectAllSlots ( ) [private]
```

Connect all slots to their signals.

#### 14.21.3.22 disableFileFolderPrompt

```
void MainWindow::disableFileFolderPrompt ( ) [slot]
```

Hide new folder edit for files and cancel button.

#### 14.21.3.23 disableMediaFolderPrompt

```
void MainWindow::disableMediaFolderPrompt ( ) [slot]
```

Hide new folder edit for media and cancel button.

**14.21.3.24 disableNoteFolderPrompt**

```
void MainWindow::disableNoteFolderPrompt ( ) [slot]
```

Hide new folder edit for notes and cancel button.

**14.21.3.25 drawCloseContainer()**

```
void MainWindow::drawCloseContainer (
    const std::vector< Relation * > & container,
    qreal sizeX,
    qreal sizeY,
    qreal topLeftX,
    qreal topLeftY,
    qreal lineX1,
    qreal lineY1,
    qreal lineX2,
    qreal lineY2,
    size_t partner,
    size_t mother,
    size_t father,
    int gap,
    bool movex,
    bool movey ) [private]
```

Draw generalized cell, line and all inside person items in close family view. Like this:

```
<-----sizeX----->

topLeftX & topLeftY
v
+-----cell-----+ ^
|gap|box|gap|box|gap| sizeY
+-----+ v
      | line to middle box
```

**Parameters**

|                  |   |
|------------------|---|
| <i>container</i> | Container of persons which will be drawn. |
| <i>sizeX</i>     | Size of the cell in the x axis.           |
| <i>sizeY</i>     | Size of the cell in the y axis.           |
| <i>topLeftX</i>  | Top left point of the cell in the x axis. |
| <i>topLeftY</i>  | Top left point of the cell in the y axis. |
| <i>lineX1</i>    | Starting point of the line in x axis.     |
| <i>lineY1</i>    | Starting point of the line in y axis.     |
| <i>lineX2</i>    | Ending point of the line in x axis.       |
| <i>lineY2</i>    | Ending point of the line in y axis.       |
| <i>partner</i>   | Index of partner to use special color.    |
| <i>mother</i>    | Index of mother to use special color.     |
| <i>father</i>    | Index of father to use special color.     |
| <i>gap</i>       | Gap which is between boxes in cell.       |
| <i>movex</i>     | Whether to move boxes on the x line.      |
| <i>movey</i>     | Whether to move boxes on the y line.      |

### 14.21.3.26 drawCloseFamily()

```
void MainWindow::drawCloseFamily ( ) [private]
```

Draw close family to the graphics.

```
+-----cell-----+
|gap|box|gap|box|gap|
+-----+
+---+<-cell |
|gap|      +-----+ <-- box
|box|----|proband|
|gap|      +-----+
+---+      |
+-----cell-----+
|gap|box|gap|box|gap|
+-----+
```

### 14.21.3.27 drawFamilyTree()

```
void MainWindow::drawFamilyTree ( ) [private]
```

Draw family tree to the corresponding person.

```
      ....      ....
      \ /      \ /
+--box--+ +--box--+
|father | |mother |
+-----+ +-----+
      \ /      / \
      +--box--+
      |proband|
      +-----+
      |      |
+--box--+ +--box--+
|child  | |child  |
+-----+ +-----+
      /|\      |
      ...
```

### 14.21.3.28 drawGeneralGenerations()

```
void MainWindow::drawGeneralGenerations (
    Person * p,
    qreal width,
    qreal midx,
    qreal midy,
    int horizontalGap,
    int verticalGap,
    size_t barrier,
    const std::vector< size_t > & container,
    bool up,
    std::function< std::pair< size_t, size_t >(Person *, size_t, size_t)> number←
Func,
    std::function< void(Person *, qreal, qreal, qreal, int, int, size_t)> drawFunc )
[private]
```

Draw either top or bottom generation in tree.

#### Parameters

|              |   |
|--------------|---|
| <i>p</i>     | Current person.                                 |
| <i>width</i> | Width of the place where will be the relations. |



## Parameters

|                      |  |
|----------------------|--|
| <i>midx</i>          | What is the middle in the x axis of the place.             |
| <i>midy</i>          | What is the middle in the y axis of the place.             |
| <i>horizontalGap</i> | Min gap between boxes in the x axis.                       |
| <i>verticalGap</i>   | Gap between generations.                                   |
| <i>barrier</i>       | How more generations to show.                              |
| <i>container</i>     | Vector of relations to draw.                               |
| <i>up</i>            | If we are drawing upwards or downwards.                    |
| <i>numberFunc</i>    | Function for computing the widths and depths of the trees. |
| <i>drawFunc</i>      | Function to recursively call for drawing.                  |

**14.21.3.29 drawOlderGenerations()**

```
void MainWindow::drawOlderGenerations (
    Person * p,
    qreal width,
    qreal midx,
    qreal midy,
    int horizontalGap,
    int verticalGap,
    size_t barrier ) [private]
```

Draw one family tree box and recursively make other.

```
+-----+
|<-xfrom ....      | Box will be in the middle.
|<-yfrom ....      | Then it will call recursively on the top half the space.
+-----+
<-----space----->
```

## Parameters

|                      |   |
|----------------------|---|
| <i>p</i>             | Which person we are drawing.                        |
| <i>width</i>         | What is the x size where should the box take place. |
| <i>midx</i>          | What is the middle in the x axis of the place.      |
| <i>midy</i>          | What is the middle in the y axis of the place.      |
| <i>horizontalGap</i> | Min gap between boxes in the x axis.                |
| <i>verticalGap</i>   | Gap between generations.                            |
| <i>barrier</i>       | How more generations to show.                       |

**14.21.3.30 drawPartner()**

```
void MainWindow::drawPartner (
    qreal canvasMidX,
    qreal canvasMidY,
    int horizontalGap ) [private]
```

Draw partner (to the left of the main person) to the family tree if there is one.

## Parameters

|                      |                               |
|----------------------|-------------------------------|
| <i>canvasMidX</i>    | Centre-point of the canvas.   |
| <i>canvasMidY</i>    | Centre-point of the canvas.   |
| <i>horizontalGap</i> | Gap in between persons boxes. |

**14.21.3.31 drawYoungerGenerations()**

```
void MainWindow::drawYoungerGenerations (
    Person * p,
    qreal width,
    qreal midx,
    qreal midy,
    int horizontalGap,
    int verticalGap,
    size_t barrier ) [private]
```

Draw all younger relations to the tree diagram.

```
+---+      +---+      +---+
|box|      |box|      |box|
+---+      +---+      +---+
.          /  \        .
..... +---+ +---+ .....
.      |box| |box| .
..... +---+ +---+ .....
```

## Parameters

|                      |   |
|----------------------|---|
| <i>p</i>             | Current person which is above.                              |
| <i>width</i>         | Width of the place where will be the children drawn.        |
| <i>midx</i>          | What is the middle in the x axis of the place for children. |
| <i>midy</i>          | What is the middle in the y axis of the place for children. |
| <i>horizontalGap</i> | Min gap between boxes in the x axis.                        |
| <i>verticalGap</i>   | Gap between generations.                                    |
| <i>barrier</i>       | How more generations to show.                               |

**14.21.3.32 editEvent [1/2]**

```
void MainWindow::editEvent (
    QTreeWidgetItem * item ) [slot]
```

Edit event when clicking on it by opening a dialog window.

## Parameters

|             |                         |
|-------------|-------------------------|
| <i>item</i> | Which item was clicked. |
|-------------|-------------------------|

**14.21.3.33 editEvent() [2/2]**

```
void MainWindow::editEvent (
```

```
size_t id ) [private]
```

Edit event by its id.

#### Parameters

|           |                  |
|-----------|------------------|
| <i>id</i> | Id of the event. |
|-----------|------------------|

### 14.21.3.34 editRelation [1/2]

```
void MainWindow::editRelation (
    QTreeWidgetItem * item ) [slot]
```

Edit relation when clicking on it by opening a dialog window.

#### Parameters

|             |                         |
|-------------|-------------------------|
| <i>item</i> | Which item was clicked. |
|-------------|-------------------------|

### 14.21.3.35 editRelation() [2/2]

```
void MainWindow::editRelation (
    size_t id ) [private]
```

Edit relation by its id.

#### Parameters

|           |                     |
|-----------|---------------------|
| <i>id</i> | Id of the relation. |
|-----------|---------------------|

### 14.21.3.36 editTag

```
void MainWindow::editTag (
    QTreeWidgetItem * item = nullptr ) [slot]
```

Edit tag when clicking on it or creating new one.

#### Parameters

|             |                                  |
|-------------|----------------------------------|
| <i>item</i> | Which item was selected, if any. |
|-------------|----------------------------------|

### 14.21.3.37 enableFileFolderPrompt

```
void MainWindow::enableFileFolderPrompt ( ) [slot]
```

Show new folder edit for files and cancel button.

#### 14.21.3.38 enableGeneralFolderPrompt()

```
void MainWindow::enableGeneralFolderPrompt (
    bool visible,
    FileType type ) [private]
```

Show or hide new folder prompt.

##### Parameters

|                |  |
|----------------|--|
| <i>visible</i> | If the prompt should be visible or not.          |
| <i>type</i>    | For which type we are showing/hiding the prompt. |

#### 14.21.3.39 enableGeneralFolderRenamePrompt()

```
void MainWindow::enableGeneralFolderRenamePrompt (
    FileType type ) [private]
```

Show rename file prompt.

##### Parameters

|             |                            |
|-------------|----------------------------|
| <i>type</i> | Which type is to be shown. |
|-------------|----------------------------|

#### 14.21.3.40 enableMediaFolderPrompt

```
void MainWindow::enableMediaFolderPrompt ( ) [slot]
```

Show new folder edit for media and cancel button.

#### 14.21.3.41 enableNoteFolderPrompt

```
void MainWindow::enableNoteFolderPrompt ( ) [slot]
```

Show new folder edit for notes and cancel button.

#### 14.21.3.42 exportCloseFamilyDiagram()

```
bool MainWindow::exportCloseFamilyDiagram (
    double scale,
    const std::string & filename = EMPTY_STRING )
```

Export close family diagram.

##### Parameters

|                 |   |
|-----------------|---|
| <i>scale</i>    | What is the scale for this picture.   |
| <i>filename</i> | Where the picture should be saved, if it is empty prompt a dialog to choose file. |

**Returns**

If the export was successful or not.

**14.21.3.43 exportCloseFamilyDiagramDialog**

```
void MainWindow::exportCloseFamilyDiagramDialog ( ) [slot]
```

Open dialog for exporting close family diagram.

**14.21.3.44 exportGeneralDiagram()**

```
bool MainWindow::exportGeneralDiagram (
    const QString & filename,
    QGraphicsScene * gscene,
    double scale = 1 ) [private]
```

Export general picture of a diagram.

**Parameters**

|                 |                               |
|-----------------|-------------------------------|
| <i>filename</i> | What is the output file path. |
| <i>gscene</i>   | Which scene to use.           |
| <i>scale</i>    | That is set for this picture. |

**Returns**

If the export was successful or not.

**14.21.3.45 exportHtml**

```
void MainWindow::exportHtml ( ) [slot]
```

Export whole HTML project to an output directory by opening a dialog.

**14.21.3.46 exportMainHtml**

```
void MainWindow::exportMainHtml ( ) [slot]
```

Export only main person html by opening a dialog.

**14.21.3.47 exportProject**

```
void MainWindow::exportProject ( ) [slot]
```

Open dialog for exporting project.

#### 14.21.3.48 exportTemplates

```
void MainWindow::exportTemplates ( ) [slot]
```

Open dialog for exporting templates.

#### 14.21.3.49 exportTreeDiagram()

```
bool MainWindow::exportTreeDiagram (
    double scale,
    const std::string & filename = EMPTY_STRING )
```

Export tree diagram.

##### Parameters

|                 |   |
|-----------------|---|
| <i>scale</i>    | What is the scale for this picture.   |
| <i>filename</i> | Where the picture should be saved, if it is empty prompt a dialog to choose file. |

##### Returns

If the export was successful or not.

#### 14.21.3.50 exportTreeDiagramDialog

```
void MainWindow::exportTreeDiagramDialog ( ) [slot]
```

Open dialog for exporting family tree diagram.

#### 14.21.3.51 filterProjectItems

```
void MainWindow::filterProjectItems ( ) [slot]
```

Filter only matching items in project.

#### 14.21.3.52 findPerson

```
void MainWindow::findPerson ( ) [slot]
```

Focus find in project view.

#### 14.21.3.53 fitTreeInView

```
void MainWindow::fitTreeInView ( ) [slot]
```

Fit the tree view inside.

**14.21.3.54 generalGenerationNumber()**

```
std::pair< size_t, size_t > MainWindow::generalGenerationNumber (
    Person * p,
    size_t soFar,
    size_t barrier,
    const std::vector< size_t > & container,
    bool up,
    std::function< std::pair< size_t, size_t > (Person *, size_t, size_t)> func )
[private]
```

Compute recursively the maximum number of younger or older generations for the given person.

**Parameters**

|                  |   |
|------------------|---|
| <i>p</i>         | Given person.                           |
| <i>soFar</i>     | What is the depth so far.               |
| <i>barrier</i>   | How many more generations to show.      |
| <i>container</i> | Vector of relations in given direction. |
| <i>up</i>        | If the direction is up or not.          |
| <i>func</i>      | Which function to recursively call.     |

**Returns**

First is what is the width and second is the depth of the tree.

**14.21.3.55 importProject**

```
void MainWindow::importProject ( ) [slot]
```

Choose file to import from other project.

**14.21.3.56 importRootDir**

```
void MainWindow::importRootDir ( ) [slot]
```

Choose root directory of other project to import from.

**14.21.3.57 importTemplates**

```
void MainWindow::importTemplates ( ) [slot]
```

Choose file to import templates.

**14.21.3.58 initializeProjectSubView()**

```
void MainWindow::initializeProjectSubView (
    ProjectItem * parent,
    Person * person ) [private]
```

Initialize all sub-folder for item in project.

## Parameters

|               |                                      |
|---------------|--------------------------------------|
| <i>parent</i> | Parent item holding these sub items. |
| <i>person</i> | To which person it is bounded to.    |

**14.21.3.59 initializeProjectView()**

```
void MainWindow::initializeProjectView ( ) [private]
```

Initialize project view when loading new family tree.

**14.21.3.60 interractWithProjectItem**

```
void MainWindow::interractWithProjectItem (
    QTreeWidgetItem * item ) [slot]
```

Show the project item upon clicking it. It depends on the type of the item. For files open the file (same for notes and media) for person select it as a main. For events and relations open dialog with editing them.

## Parameters

|             |                            |
|-------------|----------------------------|
| <i>item</i> | Which item was clicked on. |
|-------------|----------------------------|

**14.21.3.61 loadSettings()**

```
void MainWindow::loadSettings ( ) [private]
```

Load the setting from configuration.

**14.21.3.62 olderGenerationNumber()**

```
std::pair< size_t, size_t > MainWindow::olderGenerationNumber (
    Person * p,
    int soFar,
    size_t barrier ) [private]
```

Get the number of upper generations for a person.

## Parameters

|                |  |
|----------------|--|
| <i>p</i>       | Which person we are looking at.            |
| <i>soFar</i>   | What is the number so far to the person p. |
| <i>barrier</i> | How more generations to show.              |



### Returns

First is what is the maximum to the width and second is how high the tree is.

#### 14.21.3.63 openBirthDateDialog

```
void MainWindow::openBirthDateDialog ( ) [slot]
```

Open dialog for custom Birth [Date](#).

#### 14.21.3.64 openCustomDateDialog()

```
void MainWindow::openCustomDateDialog (
    WrappedDate * date ) [private]
```

Open custom date dialog for a date.

### Parameters

|             |  |
|-------------|--|
| <i>date</i> | Pointer to the date, that is being edited. |
|-------------|--|

#### 14.21.3.65 openDeathDateDialog

```
void MainWindow::openDeathDateDialog ( ) [slot]
```

Open dialog for custom Death [Date](#).

#### 14.21.3.66 openEventTemplatesDialog

```
void MainWindow::openEventTemplatesDialog ( ) [slot]
```

Open event templates dialog.

#### 14.21.3.67 openExistingProject

```
void MainWindow::openExistingProject ( ) [slot]
```

Open existing root directory of a project.

#### 14.21.3.68 openExternalFile()

```
void MainWindow::openExternalFile (
    const std::string & fileURL ) [static]
```

Open general file in an external app using QDesktopService if possible otherwise opening with a command.

**Parameters**

|                |                  |
|----------------|------------------|
| <i>fileURL</i> | URL to the file. |
|----------------|------------------|

**14.21.3.69 openGeneralFile()**

```
void MainWindow::openGeneralFile (
    QTreeWidgetItem * item,
    FileType type ) [private]
```

Open file after double clicking it.

**Parameters**

|             |                               |
|-------------|-------------------------------|
| <i>item</i> | Which item was clicked.       |
| <i>type</i> | What is the type of the file. |

**14.21.3.70 openGeneralProject()**

```
void MainWindow::openGeneralProject (
    const std::string & dirPath ) [private]
```

Open general project by opening existing one from the list or new one.

**Parameters**

|                |  |
|----------------|--|
| <i>dirPath</i> | Path to root directory of the project. |
|----------------|--|

**14.21.3.71 openGraphicSettings**

```
void MainWindow::openGraphicSettings ( ) [slot]
```

Open dialog for graphic settings.

**14.21.3.72 openNewFamilyTreeProject**

```
void MainWindow::openNewFamilyTreeProject ( ) [slot]
```

Open new empty project.

**14.21.3.73 openRelationTemplatesDialog**

```
void MainWindow::openRelationTemplatesDialog ( ) [slot]
```

Open relation templates dialog.

#### 14.21.3.74 openSelectedFile

```
void MainWindow::openSelectedFile (
    QTreeWidgetItem * item ) [slot]
```

Open file after double clicking it.

##### Parameters

|             |                            |
|-------------|----------------------------|
| <i>item</i> | Which item was clicked on. |
|-------------|----------------------------|

#### 14.21.3.75 openSelectedMedia

```
void MainWindow::openSelectedMedia (
    QTreeWidgetItem * item ) [slot]
```

Open media after double clicking it.

##### Parameters

|             |                            |
|-------------|----------------------------|
| <i>item</i> | Which item was clicked on. |
|-------------|----------------------------|

#### 14.21.3.76 openSelectedNote

```
void MainWindow::openSelectedNote (
    QTreeWidgetItem * item ) [slot]
```

Open note after double clicking it.

##### Parameters

|             |                            |
|-------------|----------------------------|
| <i>item</i> | Which item was clicked on. |
|-------------|----------------------------|

#### 14.21.3.77 promoteRelation

```
void MainWindow::promoteRelation ( ) [slot]
```

Promote current relation.

#### 14.21.3.78 refreshDates

```
void MainWindow::refreshDates ( ) [slot]
```

Refresh all dates in persons info tab.

#### 14.21.3.79 refreshEventTab

```
void MainWindow::refreshEventTab ( ) [slot]
```

Refresh list with events.

#### 14.21.3.80 refreshGraphics

```
void MainWindow::refreshGraphics ( ) [slot]
```

Refresh all graphics.

#### 14.21.3.81 refreshPersonsInfoTab()

```
void MainWindow::refreshPersonsInfoTab ( ) [private]
```

Refresh basic information about main person.

#### 14.21.3.82 refreshProjectEventsView()

```
void MainWindow::refreshProjectEventsView (
    ProjectItem * item,
    Person * person ) [private]
```

Refresh project view for events.

##### Parameters

|               |   |
|---------------|---|
| <i>item</i>   | Group item for all events.              |
| <i>person</i> | For which person we are showing events. |

#### 14.21.3.83 refreshProjectFilesView()

```
void MainWindow::refreshProjectFilesView (
    ProjectItem * item,
    Person * person ) [private]
```

Refresh project view for files.

##### Parameters

|               |  |
|---------------|--|
| <i>item</i>   | Group item for all files.              |
| <i>person</i> | For which person we are showing files. |

#### 14.21.3.84 refreshProjectGeneralFilesView()

```
void MainWindow::refreshProjectGeneralFilesView (
    const QIcon & containerIcon,
    const QString & containerName,
    const QIcon & fileIcon,
    FileType type,
    ProjectItem * item,
    Person * person ) [private]
```

Refresh project view with general files.

##### Parameters

|                      |   |
|----------------------|---|
| <i>containerIcon</i> | Icon of the container having all files.       |
| <i>containerName</i> | Name of the container having all files.       |
| <i>fileIcon</i>      | Icon for the file itself.                     |
| <i>type</i>          | Type of the file.                             |
| <i>item</i>          | Which item is the container.                  |
| <i>person</i>        | To which person these containers are bounded. |

#### 14.21.3.85 refreshProjectMediaView()

```
void MainWindow::refreshProjectMediaView (
    ProjectItem * item,
    Person * person ) [private]
```

Refresh project view for media.

##### Parameters

|               |  |
|---------------|--|
| <i>item</i>   | Group item for all media.              |
| <i>person</i> | For which person we are showing media. |

#### 14.21.3.86 refreshProjectNoteView()

```
void MainWindow::refreshProjectNoteView (
    ProjectItem * item,
    Person * person ) [private]
```

Refresh project view for notes.

##### Parameters

|               |  |
|---------------|--|
| <i>item</i>   | Group item for all notes.              |
| <i>person</i> | For which person we are showing notes. |

#### 14.21.3.87 refreshProjectRelationsView()

```
void MainWindow::refreshProjectRelationsView (
    ProjectItem * item,
    Person * person ) [private]
```

Refresh project view for relations.

##### Parameters

|               |  |
|---------------|--|
| <i>item</i>   | Group item for all relations.              |
| <i>person</i> | For which person we are showing relations. |

#### 14.21.3.88 refreshProjectView

```
void MainWindow::refreshProjectView ( ) [slot]
```

Refresh the whole project that is shown.

#### 14.21.3.89 refreshRelationTab

```
void MainWindow::refreshRelationTab ( ) [slot]
```

Refresh relations lists.

#### 14.21.3.90 refreshTabsWithFiles

```
void MainWindow::refreshTabsWithFiles ( ) [slot]
```

Refresh all files tabs.

#### 14.21.3.91 refreshTagsList

```
void MainWindow::refreshTagsList ( ) [slot]
```

Refresh shown tags.

#### 14.21.3.92 refreshUi()

```
void MainWindow::refreshUi ( ) [private]
```

Refresh the whole UI (tabs, project and graphics).

#### 14.21.3.93 removeEvent

```
void MainWindow::removeEvent ( ) [slot]
```

Remove event from family tree.

**14.21.3.94 removeFileFolder**

```
void MainWindow::removeFileFolder ( ) [slot]
```

Remove file folder.

**14.21.3.95 removeGeneralFolder()**

```
void MainWindow::removeGeneralFolder (
    FileType type ) [private]
```

Remove folder from the tree widget.

**Parameters**

|             |   |
|-------------|---|
| <i>type</i> | Which type is used. To determine which tree widget is used. |
|-------------|---|

**14.21.3.96 removeMediaFolder**

```
void MainWindow::removeMediaFolder ( ) [slot]
```

Remove media folder.

**14.21.3.97 removeNoteFolder**

```
void MainWindow::removeNoteFolder ( ) [slot]
```

Remove note folder.

**14.21.3.98 removePerson**

```
void MainWindow::removePerson ( ) [slot]
```

Remove main person.

**14.21.3.99 removeRelation**

```
void MainWindow::removeRelation ( ) [slot]
```

Remove current relation.

**14.21.3.100 removeTag**

```
void MainWindow::removeTag ( ) [slot]
```

Delete current tag selected on the person.

#### 14.21.3.101 renameFile

```
void MainWindow::renameFile ( ) [slot]
```

Rename file or folder with files.

#### 14.21.3.102 renameGeneralFileItem()

```
void MainWindow::renameGeneralFileItem (
    FileType type ) [private]
```

Rename general file tree item.

##### Parameters

|             |   |
|-------------|---|
| <i>type</i> | Which type is used. To determine which tree widget is used. |
|-------------|---|

#### 14.21.3.103 renameMedia

```
void MainWindow::renameMedia ( ) [slot]
```

Rename media or folder with media.

#### 14.21.3.104 renameNote

```
void MainWindow::renameNote ( ) [slot]
```

Rename note or folder with notes.

#### 14.21.3.105 saveDatabase

```
bool MainWindow::saveDatabase ( ) [slot]
```

Save the database.

##### Returns

If the database was stored.

#### 14.21.3.106 saveGraphicsSetting

```
void MainWindow::saveGraphicsSetting (
    QColor probandColor,
    QColor promotedColor,
    QColor standardColor,
    QColor highlightedColor,
    int sizeX,
    int sizeY,
    QFont font,
    QColor lineColor,
    QColor textColor,
    int lineWidth,
    int rounding ) [slot]
```

Save graphics settings from its dialog.



## Parameters

|                         |                                    |
|-------------------------|------------------------------------|
| <i>probandColor</i>     | Color used for proband.            |
| <i>promotedColor</i>    | Color for promoted Relations.      |
| <i>standardColor</i>    | Standard color.                    |
| <i>highlightedColor</i> | Highlighted color.                 |
| <i>sizeX</i>            | Size of the box in x axis.         |
| <i>sizeY</i>            | Size of the box in y axis.         |
| <i>font</i>             | Used font for the text in the box. |
| <i>lineColor</i>        | Color for lines.                   |
| <i>textColor</i>        | Color for texts.                   |
| <i>lineWidth</i>        | Width of the lines.                |
| <i>rounding</i>         | Rounding of the boxes.             |

**14.21.3.107 savePersonsInfo()**

```
void MainWindow::savePersonsInfo ( ) [private]
```

Save all basic information about main person.

**14.21.3.108 selectOlderGenerationOnly**

```
void MainWindow::selectOlderGenerationOnly ( ) [slot]
```

De-select all other relation boxes other than older.

**14.21.3.109 selectSameGenerationOnly**

```
void MainWindow::selectSameGenerationOnly ( ) [slot]
```

De-select all other relation boxes other than same.

**14.21.3.110 selectYoungerGenerationOnly**

```
void MainWindow::selectYoungerGenerationOnly ( ) [slot]
```

De-select all other relation boxes other than younger.

**14.21.3.111 setEnabledDeathAtributes**

```
void MainWindow::setEnabledDeathAtributes (
    bool disabled ) [slot]
```

Change the visibility of all widgets related to death attributes.

## Parameters

|                 |   |
|-----------------|---|
| <i>disabled</i> | If the widgets should be disabled or not. |
|-----------------|---|

**14.21.3.112 setEventTabVisible**

```
void MainWindow::setEventTabVisible ( ) [slot]
```

Hide or show the event tab.

**14.21.3.113 setFileTabVisible**

```
void MainWindow::setFileTabVisible ( ) [slot]
```

Hide or show the file tab.

**14.21.3.114 setGeneralTabVisible**

```
void MainWindow::setGeneralTabVisible ( ) [slot]
```

Hide or show the general tab.

**14.21.3.115 setMainPerson**

```
void MainWindow::setMainPerson (
    size_t personId ) [slot]
```

Set the main person by clicking on graphics item.

## Parameters

|                 |   |
|-----------------|---|
| <i>personId</i> | <a href="#">Person</a> id in the clicked box. |
|-----------------|---|

**14.21.3.116 setMediaTabVisible**

```
void MainWindow::setMediaTabVisible ( ) [slot]
```

Hide or show the media tab.

**14.21.3.117 setNoteTabVisible**

```
void MainWindow::setNoteTabVisible ( ) [slot]
```

Hide or show the note tab.

**14.21.3.118 setNumberOfGenerations()**

```
std::pair< int, int > MainWindow::setNumberOfGenerations (
    int up,
    int down )
```

Set the number of generations to show.

**Parameters**

|             |  |
|-------------|--|
| <i>up</i>   | Number of generations above (older).   |
| <i>down</i> | Number of generations below (younger). |

**Returns**

Values that were set before (in the same order, first up then down).

**14.21.3.119 setRelationTabVisible**

```
void MainWindow::setRelationTabVisible ( ) [slot]
```

Hide or show the relation tab.

**14.21.3.120 setTabEast**

```
void MainWindow::setTabEast ( ) [slot]
```

Set the setting of the tab to east.

**14.21.3.121 setTabNorth**

```
void MainWindow::setTabNorth ( ) [slot]
```

Set the setting of the tab to north.

**14.21.3.122 setTabSouth**

```
void MainWindow::setTabSouth ( ) [slot]
```

Set the setting of the tab to south.

**14.21.3.123 setTabWest**

```
void MainWindow::setTabWest ( ) [slot]
```

Set the setting of the tab to west.

**14.21.3.124 setUiElements()**

```
void MainWindow::setUiElements ( ) [private]
```

Set all additional static widgets of the user interface.

**14.21.3.125 showGeneralFiles()**

```
void MainWindow::showGeneralFiles (
    QTreeWidgetItem * widget,
    VirtualDrive * root,
    FileType type ) [private]
```

Show files from container to its widget.

**Parameters**

|               |  |
|---------------|--|
| <i>widget</i> | QTreeWidgetItem to show all the files.         |
| <i>root</i>   | Root folder of all files.                      |
| <i>type</i>   | Which type of files are going to be displayed. |

**14.21.3.126 showHelp**

```
void MainWindow::showHelp ( ) [slot]
```

Show generic help.

**14.21.3.127 showRenameFile**

```
void MainWindow::showRenameFile ( ) [slot]
```

Show prompt for renaming file.

**14.21.3.128 showRenameMedia**

```
void MainWindow::showRenameMedia ( ) [slot]
```

Show prompt for renaming media.

**14.21.3.129 showRenameNote**

```
void MainWindow::showRenameNote ( ) [slot]
```

Show prompt for renaming note.

#### 14.21.3.130 splitterChanged

```
void MainWindow::splitterChanged ( ) [slot]
```

When the splitter changed.

#### 14.21.3.131 switchNewTreeProject

```
void MainWindow::switchNewTreeProject ( ) [slot]
```

Save tree and create new empty tree to show.

#### 14.21.3.132 youngerGenerationNumber()

```
std::pair< size_t, size_t > MainWindow::youngerGenerationNumber (
    Person * p,
    size_t soFar,
    size_t barrier ) [private]
```

Compute the maximum number of younger generations of the person.

##### Parameters

|                |                               |
|----------------|-------------------------------|
| <i>p</i>       | Given person.                 |
| <i>soFar</i>   | What is the depth so far.     |
| <i>barrier</i> | How more generations to show. |

##### Returns

First is what is the maximum to the width and second is how deep the tree is.

#### 14.21.3.133 zoomIn

```
void MainWindow::zoomIn ( ) [slot]
```

Zoom in on the family tree view.

#### 14.21.3.134 zoomOut

```
void MainWindow::zoomOut ( ) [slot]
```

Zoom out of the family tree view.

### 14.21.4 Member Data Documentation

#### 14.21.4.1 borderRadius\_

```
int MainWindow::borderRadius_ [private]
```

Radius of the boxes.

#### 14.21.4.2 font\_

```
QFont MainWindow::font_ [private]
```

Used font for view.

#### 14.21.4.3 FT

```
FamilyTree MainWindow::FT [private]
```

Family tree.

#### 14.21.4.4 genSizeDown\_

```
int MainWindow::genSizeDown_ [private]
```

How many generations to show in tree view down.

#### 14.21.4.5 genSizeUp\_

```
int MainWindow::genSizeUp_ [private]
```

How many generations to show in tree view up.

#### 14.21.4.6 highlightedColor\_

```
QColor MainWindow::highlightedColor_ [private]
```

Color for highlighted boxes.

#### 14.21.4.7 linePen\_

```
QPen MainWindow::linePen_ [private]
```

Pen used for drawing lines.

#### 14.21.4.8 probandColor\_

```
QColor MainWindow::probandColor_ [private]
```

Color of proband.

#### 14.21.4.9 promotedColor\_

```
QColor MainWindow::promotedColor_ [private]
```

Color for promoted father, mother and partner.

#### 14.21.4.10 scene

```
QGraphicsScene* MainWindow::scene [private]
```

Scene shown on the graphical view.

#### 14.21.4.11 sizeX\_

```
int MainWindow::sizeX_ [private]
```

X size of the person box.

#### 14.21.4.12 sizeY\_

```
int MainWindow::sizeY_ [private]
```

Y size of the person box.

#### 14.21.4.13 standardColor\_

```
QColor MainWindow::standardColor_ [private]
```

[Color](#) for other relations.

#### 14.21.4.14 textPen\_

```
QPen MainWindow::textPen_ [private]
```

Pen used for texts.

#### 14.21.4.15 treeScene

```
QGraphicsScene* MainWindow::treeScene [private]
```

Tree view scene;.

#### 14.21.4.16 treeView

```
GraphicsView* MainWindow::treeView [private]
```

Tree view graphics view.

#### 14.21.4.17 ui

```
Ui::MainWindow* MainWindow::ui [private]
```

Qt UI framework.

The documentation for this class was generated from the following files:

- graphics/mainwindow.h
- graphics/mainwindow.cpp
- graphics/mainwindow\_slots.cpp

## 14.22 Parser Class Reference

Main class for working with the database.

```
#include <file_parser.h>
```

### Public Member Functions

- [Parser](#) ()  
*Default constructor.*
- void [checkFileConsistency](#) (std::set< std::string > &database, std::vector< std::string > &found, const std::string &dir)
- void [clear](#) ()  
*Clear stored data.*
- bool [containsBackupFile](#) (std::string &backupFile)
- bool [copyFile](#) (const std::string &filePath, const std::string &dir, std::string &fileName, size\_t id)
- void [createEmptyFile](#) (const std::string &dir, const std::string &fileName)
- std::string [createResourcesDir](#) (const std::string &target)
- std::string [getAbsolutePath](#) (const std::string &dir, const std::string &fileName) const
- void [getFilePath](#) (const std::string &dir, const std::string &fileName, std::string &filePath) const
- bool [isRootDirectorySet](#) ()
- void [log](#) (const std::string &error)
- void [makeNewDatabase](#) (const std::string &dirPath)
- std::pair< bool, std::vector< std::string > > [readConfig](#) ()
- bool [readJSONFile](#) (const std::string &fileName, Json::Value &root, bool inDatabase=true)
- void [removeBackup](#) ()  
*Remove backup files if there is any.*
- void [removeFile](#) (const std::string &filename, const std::string &dir)
- void [renameFile](#) (const std::string &originalName, const std::string &changedName, const std::string &dir)
- void [restoreBackup](#) (const std::string &backupFile)
- bool [setDatabase](#) (const std::string &dirPath)
- void [writeConfig](#) (bool help)
- void [writeConfig](#) (const std::vector< std::string > &rootPaths, bool help)
- void [writeHtml](#) (const std::string &html, const std::string &output)
- void [writeHtml](#) (const std::string &html, const std::string &dirPath, const std::string &filename)
- void [writeJSON](#) (const std::string &data, const std::string &file, bool inDatabase=true)



## Private Member Functions

- bool [containsBackup](#) (std::string &backupFile, const std::string &filename)
- void [createAllDirs](#) ()  
*Create all (empty) directories in the root directory.*
- void [createJsonFile](#) (const std::string &fileName)

## Private Attributes

- std::filesystem::path [root\\_](#)  
*Path to the root directory.*
- std::filesystem::path [configPath\\_](#)  
*Path to the configuration file.*

### 14.22.1 Detailed Description

Main class for working with the database.

### 14.22.2 Constructor & Destructor Documentation

#### 14.22.2.1 Parser()

```
Parser::Parser ( )
```

Default constructor.

### 14.22.3 Member Function Documentation

#### 14.22.3.1 checkFileConsistency()

```
void Parser::checkFileConsistency (
    std::set< std::string > & database,
    std::vector< std::string > & found,
    const std::string & dir )
```

Check the consistency of files.

#### Parameters

|                 |   |
|-----------------|---|
| <i>database</i> | Which files are stored in database, these will be removed from the list if found. |
| <i>found</i>    | Files found and not present in database.  |
| <i>dir</i>      | Which directory to look into.   |

#### 14.22.3.2 clear()

```
void Parser::clear ( )
```

Clear stored data.

### 14.22.3.3 containsBackup()

```
bool Parser::containsBackup (
    std::string & backupFile,
    const std::string & filename ) [private]
```

If there is backup for this file-name.

#### Parameters

|                   |  |
|-------------------|--|
| <i>backupFile</i> | If this backup file exists put there the name of it. |
| <i>filename</i>   | Name of the file checked for having backup.          |

#### Returns

True if the backup exists.

### 14.22.3.4 containsBackupFile()

```
bool Parser::containsBackupFile (
    std::string & backupFile )
```

If the database store some backup files, which may be used. Note there can be only one backup file (when the app is working properly).

#### Parameters

|                   |   |
|-------------------|---|
| <i>backupFile</i> | If there is backup there will be stored the name of the file. |
|-------------------|---|

#### Returns

True if there is a backup file.

### 14.22.3.5 copyFile()

```
bool Parser::copyFile (
    const std::string & filePath,
    const std::string & dir,
    std::string & fileName,
    size_t id )
```

Create a new file by copying existing file on the disk.

#### Parameters

|                 |   |
|-----------------|---|
| <i>filePath</i> | Path to the original file on the disk.                      |
| <i>dir</i>      | Which directory to use in database of the root directory.   |
| <i>fileName</i> | Is the name of the file in directory. Use for return value. |
| <i>id</i>       | Id of the new file, to create the real name.                |

**Returns**

True if the process was successful, false otherwise.

**14.22.3.6 createAllDirs()**

```
void Parser::createAllDirs ( ) [private]
```

Create all (empty) directories in the root directory.

**14.22.3.7 createEmptyFile()**

```
void Parser::createEmptyFile (
    const std::string & dir,
    const std::string & fileName )
```

Create an empty file.

**Parameters**

|                 |  |
|-----------------|--|
| <i>dir</i>      | Which directory in database to use.          |
| <i>fileName</i> | What is the real file name in its directory. |

**14.22.3.8 createJsonFile()**

```
void Parser::createJsonFile (
    const std::string & fileName ) [private]
```

Create an empty JSON file.

**Parameters**

|                 |   |
|-----------------|---|
| <i>fileName</i> | Name of the file in the root directory. |
|-----------------|---|

**14.22.3.9 createResourcesDir()**

```
std::string Parser::createResourcesDir (
    const std::string & target )
```

Create resource file for diagrams for HTML export.

**Parameters**

|               |  |
|---------------|--|
| <i>target</i> | Where the resource directory should be (either a path to the directory or an html file). |
|---------------|--|

**Returns**

String representing the path to the resources directory.

**14.22.3.10 getAbsolutePath()**

```
std::string Parser::getAbsolutePath (
    const std::string & dir,
    const std::string & fileName ) const
```

Get the absolute path to the file in database.

**Parameters**

|                 |   |
|-----------------|---|
| <i>dir</i>      | Which directory should have the file.   |
| <i>fileName</i> | What is the file-name in its directory. |

**Returns**

String representing the absolute path.

**14.22.3.11 getFileUrl()**

```
void Parser::getFileUrl (
    const std::string & dir,
    const std::string & fileName,
    std::string & fileUrl ) const
```

Get the url to the file for opening the file. Format: ' [file://](#)[file path]'.

**Parameters**

|                 |  |
|-----------------|--|
| <i>dir</i>      | Which directory should have the file.            |
| <i>fileName</i> | What is the file-name in its directory.          |
| <i>fileUrl</i>  | String representing the URL in specified format. |

**14.22.3.12 isRootDirectorySet()**

```
bool Parser::isRootDirectorySet ( )
```

Whether the directory has been already set.

**Returns**

True if the directory was already chosen.

### 14.22.3.13 log()

```
void Parser::log (
    const std::string & error )
```

Log error to the error file.

#### Parameters

|              |                      |
|--------------|----------------------|
| <i>error</i> | What error occurred. |
|--------------|----------------------|

### 14.22.3.14 makeNewDatabase()

```
void Parser::makeNewDatabase (
    const std::string & dirPath )
```

Make new empty database. Create all directories and files.

#### Parameters

|                |                                    |
|----------------|------------------------------------|
| <i>dirPath</i> | Path to the chosen root directory. |
|----------------|------------------------------------|

### 14.22.3.15 readConfig()

```
std::pair< bool, std::vector< std::string > > Parser::readConfig ( )
```

Read config file if it exists and return vector of all saved paths.

#### Returns

If help should be shown and vector of strings representing paths to the directories.

### 14.22.3.16 readJSONFile()

```
bool Parser::readJSONFile (
    const std::string & fileName,
    Json::Value & root,
    bool inDatabase = true )
```

Load the content of JSON file to Json::Value format.

#### Parameters

|                   |   |
|-------------------|---|
| <i>fileName</i>   | <a href="#">File</a> in the root directory to be used.                  |
| <i>root</i>       | Where to store parsed data from the file.                               |
| <i>inDatabase</i> | If the file is in database or not, if it is not, then it is for import. |

**Returns**

True if the parsing was successful. False otherwise.

**14.22.3.17 removeBackup()**

```
void Parser::removeBackup ( )
```

Remove backup files if there is any.

**14.22.3.18 removeFile()**

```
void Parser::removeFile (
    const std::string & filename,
    const std::string & dir )
```

Remove file with given name in given directory.

**Parameters**

|                 |                                |
|-----------------|--------------------------------|
| <i>filename</i> | Name of the file.              |
| <i>dir</i>      | Given directory with the file. |

**14.22.3.19 renameFile()**

```
void Parser::renameFile (
    const std::string & originalName,
    const std::string & changedName,
    const std::string & dir )
```

Rename a file in its directory.

**Parameters**

|                     |                               |
|---------------------|-------------------------------|
| <i>originalName</i> | Original name of the file.    |
| <i>changedName</i>  | New changed name of the file. |
| <i>dir</i>          | Directory where the file is.  |

**14.22.3.20 restoreBackup()**

```
void Parser::restoreBackup (
    const std::string & backupFile )
```

Restore backup file.

## Parameters

|                   |  |
|-------------------|--|
| <i>backupFile</i> | Which backup file is going to be restored. |
|-------------------|--|

**14.22.3.21 setDatabase()**

```
bool Parser::setDatabase (
    const std::string & dirPath )
```

Set the root directory and check its content.

## Parameters

|                |                             |
|----------------|-----------------------------|
| <i>dirPath</i> | Path to the root directory. |
|----------------|-----------------------------|

## Returns

True if all files and directories exist.

**14.22.3.22 writeConfig() [1/2]**

```
void Parser::writeConfig (
    bool help )
```

Write only current project path to the configuration.

## Parameters

|             |   |
|-------------|---|
| <i>help</i> | If help window should be shown at the start-up. |
|-------------|---|

**14.22.3.23 writeConfig() [2/2]**

```
void Parser::writeConfig (
    const std::vector< std::string > & rootPaths,
    bool help )
```

Write given paths to the root directories to the configuration file.

## Parameters

|                  |   |
|------------------|---|
| <i>rootPaths</i> | Vector of all paths to the directories.         |
| <i>help</i>      | If help window should be shown at the start-up. |

**14.22.3.24 writeHtml()** [1/2]

```
void Parser::writeHtml (
    const std::string & html,
    const std::string & dirPath,
    const std::string & filename )
```

Write HTML text to an output file (or for CSS).

**Parameters**

|                 |                                     |
|-----------------|-------------------------------------|
| <i>html</i>     | HTML text to put in an output file. |
| <i>dirPath</i>  | Path to the root directory.         |
| <i>filename</i> | Name of the file in root directory. |

**14.22.3.25 writeHtml()** [2/2]

```
void Parser::writeHtml (
    const std::string & html,
    const std::string & output )
```

Write HTML text to an output file (or for CSS).

**Parameters**

|               |                                     |
|---------------|-------------------------------------|
| <i>html</i>   | HTML text to put in an output file. |
| <i>output</i> | Output file path.                   |

**14.22.3.26 writeJSON()**

```
void Parser::writeJSON (
    const std::string & data,
    const std::string & file,
    bool inDatabase = true )
```

Write data in JSON formatting to its file. Before create a backup and after saving delete the backup.

**Parameters**

|                   |   |
|-------------------|---|
| <i>data</i>       | Preprocessed JSON format string of data.                          |
| <i>file</i>       | The name of the JSON file in the root directory.                  |
| <i>inDatabase</i> | If the file is in database or not, if it is not it is for export. |

**14.22.4 Member Data Documentation****14.22.4.1 configPath\_**

```
std::filesystem::path Parser::configPath_ [private]
```



Path to the configuration file.

#### 14.22.4.2 root\_

```
std::filesystem::path Parser::root_ [private]
```

Path to the root directory.

The documentation for this class was generated from the following files:

- [core/file\\_parser.h](#)
- [core/file\\_parser.cpp](#)

## 14.23 Person Class Reference

One person in the family tree.

```
#include <person.h>
```

### Public Member Functions

- [Person](#) ()  
*Default constructor.*
- void [addEvent](#) (size\_t id)
- void [addRelation](#) (size\_t id)
- void [addTag](#) (const std::string &tag, const std::string &value)
- [WrappedDate](#) \* [birthDate](#) ()
- [WrappedDate](#) \* [deathDate](#) ()
- std::pair< bool, const std::string & > [existsTag](#) (const std::string &tag) const
- void [exportJson](#) (std::ostream &os, bool includingEvents, bool includingRelations)
- const std::string & [getAfterTitle](#) () const
- const [WrappedDate](#) & [getBirthDate](#) () const
- const std::string & [getBirthPlace](#) () const
- const [WrappedDate](#) & [getDeathDate](#) () const
- const std::string & [getDeathPlace](#) () const
- const std::vector< size\_t > & [getEvents](#) () const
- size\_t [getFather](#) () const
- std::string [getFullName](#) () const
- const [VirtualDrive](#) & [getFilesRoot](#) (FileType type=GENERAL\_FILE) const
- [VirtualDrive](#) \* [getFilesRootPointer](#) (FileType type)
- const std::string & [getFrontTitle](#) () const
- [Gender](#) [getGender](#) () const
- size\_t [getId](#) () const
- const std::string & [getMaidenName](#) () const
- size\_t [getMother](#) () const
- const std::string & [getName](#) () const
- size\_t [getPartner](#) () const
- const std::vector< size\_t > & [getRelations](#) () const
- const std::string & [getSurname](#) () const
- const std::vector< std::pair< std::string, std::string > > & [getTags](#) () const

- void [importJson](#) (const Json::Value &value)
- bool [isAlive](#) () const
- void [printHtml](#) (std::ostream &os)
- void [promoteRelation](#) (size\_t id, [Trait](#) type)
- void [readJson](#) (const Json::Value &value)
- void [removeEvent](#) (size\_t id)
- void [removeRelation](#) (size\_t id)
- void [removeTag](#) (const std::string &tag)
- void [setAfterTitle](#) (const std::string &title)
- void [setBirthDate](#) (int y, int m, int d)
- std::string & [setBirthPlace](#) ()
- void [setDeathDate](#) (int y, int m, int d)
- std::string & [setDeathPlace](#) ()
- void [setFather](#) (size\_t father)
- void [setFilesRoot](#) (const [VirtualDrive](#) &drive, [FileType](#) type=GENERAL\_FILE, size\_t plusIndex=0)
- void [setFrontTitle](#) (const std::string &title)
- void [setGender](#) ([Gender](#) g)
- void [setId](#) (size\_t id)
- void [setLives](#) (bool lives)
- std::string & [setMaidenName](#) ()
- void [setMother](#) (size\_t mother)
- std::string & [setName](#) ()
- void [setPartner](#) (size\_t id)
- std::string & [setSurname](#) ()
- std::string [str](#) () const
- void [updateSpecialRelation](#) (size\_t relId, [Trait](#) trait)

### Private Attributes

- [WrappedDate](#) [dateOfBirth\\_](#)  
*Date of birth.*
- [WrappedDate](#) [dateOfDeath\\_](#)  
*Date of death.*
- std::vector< size\_t > [events\\_](#)  
*All events ids (from Family Tree) of the person.*
- size\_t [father\\_](#)  
*Index of relation with fatherhood trait.*
- [Gender](#) [gender\\_](#)  
*Gender of the person.*
- size\_t [id\\_](#)  
*Id of the person.*
- bool [lives\\_](#)  
*If the person is alive.*
- std::string [maidenName\\_](#)  
*Maiden name of the person. Usually for a married woman.*
- size\_t [mother\\_](#)  
*Index of relation with motherhood trait.*
- std::string [name\\_](#)  
*Name of the person.*
- size\_t [partner\\_](#)  
*Id of the partnership relation.*
- std::string [placeOfBirth\\_](#)

- Place of birth.*
  - `std::string` [placeOfDeath\\_](#)
- Place of death.*
  - `std::vector< size_t >` [relations\\_](#)
- All relations ids (from Family Tree).*
  - `VirtualDrive` [rootFileDrive\\_](#)
- Root drive for files.*
  - `VirtualDrive` [rootMediaDrive\\_](#)
- Root drive for media.*
  - `VirtualDrive` [rootNoteDrive\\_](#)
- Root drive for notes.*
  - `std::string` [surname\\_](#)
- Surname of the person.*
  - `std::vector< std::pair< std::string, std::string > >` [tags\\_](#)
- Persons tags. First is the tag itself and second is the value.*
  - `std::string` [titleAfter\\_](#)
- Title that is after the name.*
  - `std::string` [titleInFront\\_](#)
- Title that is in front of a name.*

### 14.23.1 Detailed Description

One person in the family tree.

### 14.23.2 Constructor & Destructor Documentation

#### 14.23.2.1 Person()

```
Person::Person ( )
```

Default constructor.

### 14.23.3 Member Function Documentation

#### 14.23.3.1 addEvent()

```
void Person::addEvent (
    size_t id )
```

Add new event as its id.

Parameters

|           |         |
|-----------|---------|
| <i>id</i> | Its id. |
|-----------|---------|

### 14.23.3.2 addRelation()

```
void Person::addRelation (
    size_t id )
```

Add new general relation as its id.

#### Parameters

|           |         |
|-----------|---------|
| <i>id</i> | Its id. |
|-----------|---------|

### 14.23.3.3 addTag()

```
void Person::addTag (
    const std::string & tag,
    const std::string & value )
```

Add tag to the person.

#### Parameters

|              |                   |
|--------------|-------------------|
| <i>tag</i>   | Used tag.         |
| <i>value</i> | Value of the tag. |

### 14.23.3.4 birthDate()

```
WrappedDate * Person::birthDate ( )
```

Get the pointer to the birth date.

#### Returns

Pointer to the birth date.

### 14.23.3.5 deathDate()

```
WrappedDate * Person::deathDate ( )
```

Get the pointer to the death date.

#### Returns

Pointer to the death date.

### 14.23.3.6 existsTag()

```
std::pair< bool, const std::string & > Person::existsTag (
    const std::string & tag ) const
```

If there exists value with this tag.

## Parameters

|            |            |
|------------|------------|
| <i>tag</i> | Given tag. |
|------------|------------|

## Returns

True if it exists and then the value.

**14.23.3.7 exportJson()**

```
void Person::exportJson (
    std::ostream & os,
    bool includingEvents,
    bool includingRelations )
```

Export this person.

## Parameters

|                           |                                  |
|---------------------------|----------------------------------|
| <i>os</i>                 | Given output stream.             |
| <i>includingEvents</i>    | If events should be included.    |
| <i>includingRelations</i> | If relations should be included. |

**14.23.3.8 getAfterTitle()**

```
const std::string & Person::getAfterTitle ( ) const
```

Get the title after the name.

## Returns

Constant reference to the title.

**14.23.3.9 getBirthDate()**

```
const WrappedDate & Person::getBirthDate ( ) const
```

Get the date of birth.

## Returns

[Date](#) of birth.

#### 14.23.3.10 getBirthPlace()

```
const std::string & Person::getBirthPlace ( ) const
```

Get the string of place of birth.

##### Returns

Constant reference to the string of the place of birth.

#### 14.23.3.11 getDeathDate()

```
const WrappedDate & Person::getDeathDate ( ) const
```

Get the date of death.

##### Returns

[Date](#) of death.

#### 14.23.3.12 getDeathPlace()

```
const std::string & Person::getDeathPlace ( ) const
```

Get the string representing the place of death.

##### Returns

String representing the place of death.

#### 14.23.3.13 getEvents()

```
const std::vector< size_t > & Person::getEvents ( ) const
```

Get constant reference to the list of all events.

##### Returns

List of events ids.

#### 14.23.3.14 getFather()

```
size_t Person::getFather ( ) const
```

Get the index of fatherhood relation representing father.

##### Returns

Index of the fatherhood relation.

#### 14.23.3.15 getFilesRoot()

```
const VirtualDrive & Person::getFilesRoot (
    FileType type = GENERAL_FILE ) const
```

Get constant reference to the root drive of files.

## Parameters

|             |                     |
|-------------|---------------------|
| <i>type</i> | What type of files. |
|-------------|---------------------|

## Returns

Root drive for given file type.

**14.23.3.16 getFilesRootPointer()**

```
VirtualDrive * Person::getFilesRootPointer (
    FileType type )
```

Get the pointer to the files root drive.

## Parameters

|             |                    |
|-------------|--------------------|
| <i>type</i> | Type of the files. |
|-------------|--------------------|

## Returns

Pointer to the root drive.

**14.23.3.17 getFrontTitle()**

```
const std::string & Person::getFrontTitle ( ) const
```

Get the title in front.

## Returns

Constant reference to the title.

**14.23.3.18 getFullName()**

```
std::string Person::getFullName ( ) const
```

Get the full name of a person in order: "Surname + MaidenName + Name".

## Returns

Full name of a person.

**14.23.3.19 getGender()**

```
Gender Person::getGender ( ) const
```

Get the gender of the person.

**Returns**

[Person](#)'s gender.

**14.23.3.20 getId()**

```
size_t Person::getId ( ) const
```

Get the id of the person.

**Returns**

The id of the person.

**14.23.3.21 getMaidenName()**

```
const std::string & Person::getMaidenName ( ) const
```

Get the maiden name.

**Returns**

Constant reference to the maiden name.

**14.23.3.22 getMother()**

```
size_t Person::getMother ( ) const
```

Get the index of motherhood relation representing mother.

**Returns**

Index of the motherhood relation.

**14.23.3.23 getName()**

```
const std::string & Person::getName ( ) const
```

Get the name of the person.

**Returns**

Constant reference to the name.



**14.23.3.24 getPartner()**

```
size_t Person::getPartner ( ) const
```

Get the id of the partner.

**Returns**

Id of the partnership relation.

**14.23.3.25 getRelations()**

```
const std::vector< size_t > & Person::getRelations ( ) const
```

Get constant reference to the list of all relations.

**Returns**

List of relations ids.

**14.23.3.26 getSurname()**

```
const std::string & Person::getSurname ( ) const
```

Get the surname of the person.

**Returns**

Constant reference to the surname.

**14.23.3.27 getTags()**

```
const std::vector< std::pair< std::string, std::string > > & Person::getTags ( ) const
```

Get all tags.

**Returns**

Constant reference to the tags container.

**14.23.3.28 importJson()**

```
void Person::importJson (
    const Json::Value & value )
```

Import person from single file JSON. Some values may not be present from the import.

**Parameters**

|              |             |
|--------------|-------------|
| <i>value</i> | JSON value. |
|--------------|-------------|

**14.23.3.29 isAlive()**

```
bool Person::isAlive ( ) const
```

If the person is alive or not.

**Returns**

True if the person lives.

**14.23.3.30 printHtml()**

```
void Person::printHtml (
    std::ostream & os )
```

Print person details to an output in HTML syntax.

**Parameters**

|           |                |
|-----------|----------------|
| <i>os</i> | Output stream. |
|-----------|----------------|

**14.23.3.31 promoteRelation()**

```
void Person::promoteRelation (
    size_t id,
    Trait type )
```

Promote relation to the main one. If it is already promoted de-promote it.

**Parameters**

|             |                       |
|-------------|-----------------------|
| <i>id</i>   | Of the relation.      |
| <i>type</i> | Type of the relation. |

**14.23.3.32 readJson()**

```
void Person::readJson (
    const Json::Value & value )
```

Read data from JSON to this person.

## Parameters

|              |                                 |
|--------------|---------------------------------|
| <i>value</i> | Loaded JSON data from the file. |
|--------------|---------------------------------|

**14.23.3.33 removeEvent()**

```
void Person::removeEvent (
    size_t id )
```

Remove event if the person has it (by its id).

## Parameters

|           |         |
|-----------|---------|
| <i>id</i> | Its id. |
|-----------|---------|

**14.23.3.34 removeRelation()**

```
void Person::removeRelation (
    size_t id )
```

Remove relation if the person has it (by its id) that includes children and other special relations.

## Parameters

|           |         |
|-----------|---------|
| <i>id</i> | Its id. |
|-----------|---------|

**14.23.3.35 removeTag()**

```
void Person::removeTag (
    const std::string & tag )
```

Remove tag from the person if it exists.

## Parameters

|            |                            |
|------------|----------------------------|
| <i>tag</i> | Which tag will be removed. |
|------------|----------------------------|

**14.23.3.36 setAfterTitle()**

```
void Person::setAfterTitle (
    const std::string & title )
```

Set the title after the name.

**Parameters**

|              |                |
|--------------|----------------|
| <i>title</i> | The new title. |
|--------------|----------------|

**14.23.3.37 setBirthDate()**

```
void Person::setBirthDate (
    int y,
    int m,
    int d )
```

Set the exact birth date.

**Parameters**

|          |                  |
|----------|------------------|
| <i>d</i> | Number of day.   |
| <i>m</i> | Number of month. |
| <i>y</i> | Number of year.  |

**14.23.3.38 setBirthPlace()**

```
std::string & Person::setBirthPlace ( )
```

Set the birth place by giving reference to the string.

**Returns**

Reference to the birth place.

**14.23.3.39 setDeathDate()**

```
void Person::setDeathDate (
    int y,
    int m,
    int d )
```

Set the exact death date.

**Parameters**

|          |                  |
|----------|------------------|
| <i>d</i> | Number of day.   |
| <i>m</i> | Number of month. |
| <i>y</i> | Number of year.  |

**14.23.3.40 setDeathPlace()**

```
std::string & Person::setDeathPlace ( )
```

Set the death place by giving reference to the string.

**Returns**

Reference to the death place.

**14.23.3.41 setFather()**

```
void Person::setFather (
    size_t father )
```

Set father for the person. If father is present put him into relations.

**Parameters**

|               |                               |
|---------------|-------------------------------|
| <i>father</i> | Index to fatherhood relation. |
|---------------|-------------------------------|

**14.23.3.42 setFilesRoot()**

```
void Person::setFilesRoot (
    const VirtualDrive & drive,
    FileType type = GENERAL_FILE,
    size_t plusIndex = 0 )
```

Set the drive by copying the data from the other.

**Parameters**

|                  |  |
|------------------|--|
| <i>drive</i>     | Other drive.                             |
| <i>type</i>      | What type of file is the drive.          |
| <i>plusIndex</i> | What should be added to each file index. |

**14.23.3.43 setFrontTitle()**

```
void Person::setFrontTitle (
    const std::string & title )
```

Set the title in front of a name.

**Parameters**

|              |                |
|--------------|----------------|
| <i>title</i> | The new title. |
|--------------|----------------|

#### 14.23.3.44 setGender()

```
void Person::setGender (
    Gender g )
```

Set the gender.

##### Parameters

|          |               |
|----------|---------------|
| <i>g</i> | Given gender. |
|----------|---------------|

#### 14.23.3.45 setId()

```
void Person::setId (
    size_t id )
```

Set the id of the person.

##### Parameters

|           |             |
|-----------|-------------|
| <i>id</i> | The new id. |
|-----------|-------------|

#### 14.23.3.46 setLives()

```
void Person::setLives (
    bool lives )
```

Set if the person lives or not.

##### Parameters

|              |                                    |
|--------------|------------------------------------|
| <i>lives</i> | True if the person is still alive. |
|--------------|------------------------------------|

#### 14.23.3.47 setMaidenName()

```
std::string & Person::setMaidenName ( )
```

Set the maiden name by getting the reference to it.

##### Returns

The reference to the maiden name.

#### 14.23.3.48 setMother()

```
void Person::setMother (
    size_t mother )
```

Set mother for the person. If mother is present put her into relations.

**Parameters**

|               |                               |
|---------------|-------------------------------|
| <i>mother</i> | Index to motherhood relation. |
|---------------|-------------------------------|

**14.23.3.49 setName()**

```
std::string & Person::setName ( )
```

Get the reference to the name for changing the name.

**Returns**

The reference to the name string.

**14.23.3.50 setPartner()**

```
void Person::setPartner (
    size_t id )
```

Set the partnership relation id.

**Parameters**

|           |                                 |
|-----------|---------------------------------|
| <i>id</i> | Id of the partnership relation. |
|-----------|---------------------------------|

**14.23.3.51 setSurname()**

```
std::string & Person::setSurname ( )
```

Get the reference to the surname for changing it.

**Returns**

The reference to the surname string.

**14.23.3.52 str()**

```
std::string Person::str ( ) const
```

Get the string representation of person.

**Returns**

String representing a person.

#### 14.23.3.53 updateSpecialRelation()

```
void Person::updateSpecialRelation (
    size_t relId,
    Trait trait )
```

Update relation if it is special.



## Parameters

|                    |                         |
|--------------------|-------------------------|
| <i>rel↔<br/>Id</i> | Which id it has.        |
| <i>trait</i>       | Which trait it now has. |

## 14.23.4 Member Data Documentation

### 14.23.4.1 dateOfBirth\_

`WrappedDate` `Person::dateOfBirth_` [private]

`Date` of birth.

### 14.23.4.2 dateOfDeath\_

`WrappedDate` `Person::dateOfDeath_` [private]

`Date` of death.

### 14.23.4.3 events\_

`std::vector<size_t>` `Person::events_` [private]

All events ids (from Family Tree) of the person.

### 14.23.4.4 father\_

`size_t` `Person::father_` [private]

Index of relation with fatherhood trait.

### 14.23.4.5 gender\_

`Gender` `Person::gender_` [private]

Gender of the person.

### 14.23.4.6 id\_

`size_t` `Person::id_` [private]

Id of the person.

#### 14.23.4.7 lives\_

```
bool Person::lives_ [private]
```

If the person is alive.

#### 14.23.4.8 maidenName\_

```
std::string Person::maidenName_ [private]
```

Maiden name of the person. Usually for a married woman.

#### 14.23.4.9 mother\_

```
size_t Person::mother_ [private]
```

Index of relation with motherhood trait.

#### 14.23.4.10 name\_

```
std::string Person::name_ [private]
```

Name of the person.

#### 14.23.4.11 partner\_

```
size_t Person::partner_ [private]
```

Id of the partnership relation.

#### 14.23.4.12 placeOfBirth\_

```
std::string Person::placeOfBirth_ [private]
```

Place of birth.

#### 14.23.4.13 placeOfDeath\_

```
std::string Person::placeOfDeath_ [private]
```

Place of death.

#### 14.23.4.14 relations\_

```
std::vector<size_t> Person::relations_ [private]
```

All relations ids (from Family Tree).

**14.23.4.15 rootFileDrive\_**

```
VirtualDrive Person::rootFileDrive_ [private]
```

Root drive for files.

**14.23.4.16 rootMediaDrive\_**

```
VirtualDrive Person::rootMediaDrive_ [private]
```

Root drive for media.

**14.23.4.17 rootNoteDrive\_**

```
VirtualDrive Person::rootNoteDrive_ [private]
```

Root drive for notes.

**14.23.4.18 surname\_**

```
std::string Person::surname_ [private]
```

Surname of the person.

**14.23.4.19 tags\_**

```
std::vector<std::pair<std::string, std::string> > Person::tags_ [private]
```

Persons tags. First is the tag itself and second is the value.

**14.23.4.20 titleAfter\_**

```
std::string Person::titleAfter_ [private]
```

Title that is after the name.

**14.23.4.21 titleInFront\_**

```
std::string Person::titleInFront_ [private]
```

Title that is in front of a name.

The documentation for this class was generated from the following files:

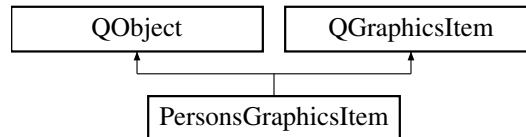
- [core/person.h](#)
- [core/person.cpp](#)

## 14.24 PersonsGraphicsItem Class Reference

Box with persons information.

```
#include <graphics_items.h>
```

Inheritance diagram for PersonsGraphicsItem:



### Signals

- void [itemClicked](#) (size\_t personId)  
*Signal when the item is pressed.*

### Public Member Functions

- [PersonsGraphicsItem](#) (const QStringList &lines, int sizeX, int sizeY, [Person](#) \*p, qreal x, qreal y, QColor color, QColor highlighted, QFont font, QWidget \*parent, QPen \*pen, QPen \*textPen, int borderRadius)
- [~PersonsGraphicsItem](#) ()  
*Default destructor.*
- QRectF [boundingRect](#) () const override
- void [paint](#) (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget) override
- QPainterPath [shape](#) () const override

### Protected Member Functions

- void [hoverEnterEvent](#) (QGraphicsSceneHoverEvent \*event) override
- void [hoverLeaveEvent](#) (QGraphicsSceneHoverEvent \*event) override
- void [mousePressEvent](#) (QGraphicsSceneMouseEvent \*event) override

### Private Attributes

- int [borderRadius\\_](#)  
*Border radius of the box.*
- int [bold\\_text\\_](#)  
*Which text should be bold.*
- QColor [color\\_](#)  
*Which color will be used to fill the box.*
- QFont [font\\_](#)  
*Which font is used.*
- QColor [highlighted\\_](#)  
*Which color will be used when highlighted.*
- bool [hovered\\_](#)  
*If the box is hovered.*

- QPen \* [pen\\_](#)  
*Pointer to the used pen.*
- [Person](#) \* [person\\_](#)  
*Given person represented in this box.*
- int [sizeX\\_](#)  
*Size of the cell in the x axis.*
- int [sizeY\\_](#)  
*Size of the cell in the y axis.*
- QStringList [textLines\\_](#)  
*All text lines to show in the text box.*
- QPen \* [textPen\\_](#)  
*Pointer to the text pen.*
- qreal [x\\_](#)  
*Top left point x position.*
- qreal [y\\_](#)  
*Top left point y position.*

### 14.24.1 Detailed Description

Box with persons information.

### 14.24.2 Constructor & Destructor Documentation

#### 14.24.2.1 PersonsGraphicsItem()

```
PersonsGraphicsItem::PersonsGraphicsItem (
    const QStringList & lines,
    int sizeX,
    int sizeY,
    Person * p,
    qreal x,
    qreal y,
    QColor color,
    QColor highlighted,
    QFont font,
    QWidget * parent,
    QPen * pen,
    QPen * textPen,
    int borderRadius )
```

Default constructor.

#### Parameters

|              |  |
|--------------|--|
| <i>lines</i> | Each separate line of text to show in the box. |
| <i>sizeX</i> | Size of the cell in the x axis.                |
| <i>sizeY</i> | Size of the cell in the y axis.                |
| <i>p</i>     | Given person represented in this box.          |
| <i>x</i>     | Top left point x position.                     |
| <i>y</i>     | Top left point y position.                     |
| <i>color</i> | Which color will be used to fill the box.      |

**Parameters**

|                     |   |
|---------------------|---|
| <i>highlighted</i>  | Which color will be used when the box is highlighted. |
| <i>font</i>         | Which font will be used to type the words.            |
| <i>parent</i>       | Which widget will take care of clicking the box.      |
| <i>pen</i>          | Which pen is used for lines.                          |
| <i>textPen</i>      | Which pen is used for text.                           |
| <i>borderRadius</i> | What is the radius of the border.                     |

**14.24.2.2 ~PersonsGraphicsItem()**

```
PersonsGraphicsItem::~PersonsGraphicsItem ( )
```

Default destructor.

**14.24.3 Member Function Documentation****14.24.3.1 boundingRect()**

```
QRectF PersonsGraphicsItem::boundingRect ( ) const [override]
```

Get the bounding rectangle of the object.

**Returns**

Rectangle that is bounding the item.

**14.24.3.2 hoverEnterEvent()**

```
void PersonsGraphicsItem::hoverEnterEvent (
    QGraphicsSceneHoverEvent * event ) [override], [protected]
```

When the box is hovered by a mouse highlight the box.

**Parameters**

|              |  |
|--------------|--|
| <i>event</i> | <a href="#">Event</a> when the item was hovered. |
|--------------|--|

**14.24.3.3 hoverLeaveEvent()**

```
void PersonsGraphicsItem::hoverLeaveEvent (
    QGraphicsSceneHoverEvent * event ) [override], [protected]
```

When the box is not hovered anymore by a mouse stop highlighting.

## Parameters

|              |   |
|--------------|---|
| <i>event</i> | Event when the item is not hovered anymore. |
|--------------|---|

**14.24.3.4 itemClicked**

```
void PersonsGraphicsItem::itemClicked (
    size_t personId ) [signal]
```

Signal when the item is pressed.

**14.24.3.5 mousePressEvent()**

```
void PersonsGraphicsItem::mousePressEvent (
    QGraphicsSceneMouseEvent * event ) [override], [protected]
```

Press event after clicking the object.

## Parameters

|              |                                  |
|--------------|----------------------------------|
| <i>event</i> | Event when the item was clicked. |
|--------------|----------------------------------|

**14.24.3.6 paint()**

```
void PersonsGraphicsItem::paint (
    QPainter * painter,
    const QStyleOptionGraphicsItem * option,
    QWidget * widget ) [override]
```

Paint the object to its painter.

## Parameters

|                |   |
|----------------|---|
| <i>painter</i> | Which painter to use to paint the object. |
| <i>option</i>  | Option for the Qt framework (not used).   |
| <i>widget</i>  | Widget for the Qt framework (not used).   |

**14.24.3.7 shape()**

```
QPainterPath PersonsGraphicsItem::shape ( ) const [override]
```

Get the shape of the item.

## Returns

Get the shape to paint.

## 14.24.4 Member Data Documentation

### 14.24.4.1 bold\_text\_

```
int PersonsGraphicsItem::bold_text_ [private]
```

Which text should be bold.

### 14.24.4.2 borderRadius\_

```
int PersonsGraphicsItem::borderRadius_ [private]
```

Border radius of the box.

### 14.24.4.3 color\_

```
QColor PersonsGraphicsItem::color_ [private]
```

Which color will be used to fill the box.

### 14.24.4.4 font\_

```
QFont PersonsGraphicsItem::font_ [private]
```

Which font is used.

### 14.24.4.5 highlighted\_

```
QColor PersonsGraphicsItem::highlighted_ [private]
```

Which color will be used when highlighted.

### 14.24.4.6 hovered\_

```
bool PersonsGraphicsItem::hovered_ [private]
```

If the box is hovered.

### 14.24.4.7 pen\_

```
QPen* PersonsGraphicsItem::pen_ [private]
```

Pointer to the used pen.



#### 14.24.4.8 person\_

```
Person* PersonsGraphicsItem::person_ [private]
```

Given person represented in this box.

#### 14.24.4.9 sizex\_

```
int PersonsGraphicsItem::sizex_ [private]
```

Size of the cell in the x axis.

#### 14.24.4.10 sizey\_

```
int PersonsGraphicsItem::sizey_ [private]
```

Size of the cell in the y axis.

#### 14.24.4.11 textLines\_

```
QStringList PersonsGraphicsItem::textLines_ [private]
```

All text lines to show in the text box.

#### 14.24.4.12 textPen\_

```
QPen* PersonsGraphicsItem::textPen_ [private]
```

Pointer to the text pen.

#### 14.24.4.13 x\_

```
qreal PersonsGraphicsItem::x_ [private]
```

Top left point x position.

#### 14.24.4.14 y\_

```
qreal PersonsGraphicsItem::y_ [private]
```

Top left point y position.

The documentation for this class was generated from the following files:

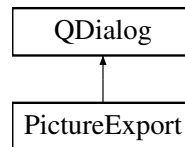
- [graphics/graphics\\_items.h](#)
- [graphics/graphics\\_items.cpp](#)

## 14.25 PictureExport Class Reference

Dialog for exporting diagram. To choose zoom of the picture.

```
#include <dialogs.h>
```

Inheritance diagram for PictureExport:



### Public Slots

- void `exportDiagram` ()  
*When OK is pressed export the diagram.*

### Public Member Functions

- `PictureExport` (bool `closeDiagram`, QWidget \*`parent`=nullptr)
- `~PictureExport` ()  
*Default destructor.*

### Private Attributes

- bool `closeDiagram_`  
*Which diagram is being exported.*
- Ui::PictureExport \* `ui`  
*Used Qt user interface.*

### 14.25.1 Detailed Description

Dialog for exporting diagram. To choose zoom of the picture.

### 14.25.2 Constructor & Destructor Documentation

#### 14.25.2.1 PictureExport()

```
PictureExport::PictureExport (
    bool closeDiagram,
    QWidget * parent = nullptr ) [explicit]
```

Default constructor.

## Parameters

|                     |   |
|---------------------|---|
| <i>closeDiagram</i> | If the diagram should be close family diagram. Otherwise it is family tree. |
| <i>parent</i>       | The Qt Widget parent.   |

**14.25.2.2 ~PictureExport()**

```
PictureExport::~PictureExport ( )
```

Default destructor.

**14.25.3 Member Function Documentation****14.25.3.1 exportDiagram**

```
void PictureExport::exportDiagram ( ) [slot]
```

When OK is pressed export the diagram.

**14.25.4 Member Data Documentation****14.25.4.1 closeDiagram\_**

```
bool PictureExport::closeDiagram_ [private]
```

Which diagram is being exported.

**14.25.4.2 ui**

```
Ui::PictureExport* PictureExport::ui [private]
```

Used Qt user interface.

The documentation for this class was generated from the following files:

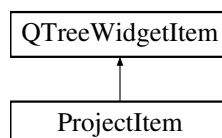
- [graphics/dialogs.h](#)
- [graphics/dialogs.cpp](#)

**14.26 ProjectItem Class Reference**

This is a tree widget item to be shown on a project view.

```
#include <graphics_items.h>
```

Inheritance diagram for ProjectItem:



## Public Member Functions

- [ProjectItem](#) ([ProjectItemType](#) type, size\_t id=0)
- size\_t [getId](#) () const
- [ProjectItemType](#) type () const

## Private Attributes

- size\_t [id\\_](#)  
*Id of the stored object.*
- [ProjectItemType](#) [type\\_](#)  
*Type of the stored object.*

## 14.26.1 Detailed Description

This is a tree widget item to be shown on a project view.

## 14.26.2 Constructor & Destructor Documentation

### 14.26.2.1 ProjectItem()

```
ProjectItem::ProjectItem (  
    ProjectItemType type,  
    size_t id = 0 ) [explicit]
```

Default constructor.

#### Parameters

|             |                                |
|-------------|--------------------------------|
| <i>type</i> | What is the type of this item. |
| <i>id</i>   | Given id of the item.          |

## 14.26.3 Member Function Documentation

### 14.26.3.1 getId()

```
size_t ProjectItem::getId ( ) const
```

Get the id of the object.

#### Returns

Id of the object.

### 14.26.3.2 type()

```
ProjectItemType ProjectItem::type ( ) const
```

Get the type of the item.

Returns

Type of the project.

## 14.26.4 Member Data Documentation

### 14.26.4.1 id\_

```
size_t ProjectItem::id_ [private]
```

Id of the stored object.

### 14.26.4.2 type\_

```
ProjectItemType ProjectItem::type_ [private]
```

Type of the stored object.

The documentation for this class was generated from the following files:

- [graphics/graphics\\_items.h](#)
- [graphics/graphics\\_items.cpp](#)

## 14.27 Relation Class Reference

Class holding data for a relationship.

```
#include <family_tree_items.h>
```

### Public Member Functions

- [Relation](#) ([Settings](#) \*settings)
- [size\\_t](#) [getFirstPerson](#) () const
- [size\\_t](#) [getId](#) () const
- [size\\_t](#) [getSecondPerson](#) () const
- [size\\_t](#) [getTemplate](#) () const
- const std::string & [getTitle](#) () const
- [size\\_t](#) [getTheOtherPerson](#) (size\_t current)
- const std::string & [getTheOtherPersonName](#) (size\_t current)
- [Trait](#) [getTrait](#) ()
- bool [isOlderGeneration](#) (size\_t id)
- bool [isParent](#) (size\_t personId)
- bool [isSameGenration](#) ()
- bool [isYoungerGeneration](#) (size\_t id)
- void [printHtml](#) (std::ostream &os, const std::string &firstPerson, const std::string &secondPerson, const std::string &promoted)
- void [readJson](#) (const Json::Value &value)
- void [setId](#) (size\_t id)
- void [setPersons](#) (size\_t person1, size\_t person2)
- void [setTemplate](#) (size\_t templ)
- std::string [str](#) (const std::string &firstPerson, const std::string &secondPerson) const

## Private Attributes

- `size_t id_`  
*Id of this relation.*
- `size_t person1_`  
*First person of the relation.*
- `size_t person2_`  
*Second person of the relation.*
- `Settings * settings_`  
*Used settings holding the templates.*
- `size_t template_`  
*Relation template it is using.*

## 14.27.1 Detailed Description

Class holding data for a relationship.

## 14.27.2 Constructor & Destructor Documentation

### 14.27.2.1 Relation()

```
Relation::Relation (
    Settings * settings ) [explicit]
```

Default constructor.

#### Parameters

|                       |                                     |
|-----------------------|-------------------------------------|
| <code>settings</code> | Pointer to the settings of the app. |
|-----------------------|-------------------------------------|

## 14.27.3 Member Function Documentation

### 14.27.3.1 getFirstPerson()

```
size_t Relation::getFirstPerson ( ) const
```

Get the first person of the relation.

#### Returns

Id of the first person.

### 14.27.3.2 getId()

```
size_t Relation::getId ( ) const
```

Get the id of the relation.

#### Returns

Id of the relation.

### 14.27.3.3 getSecondPerson()

```
size_t Relation::getSecondPerson ( ) const
```

Get the second person of the relation.

#### Returns

Id of the second person.

### 14.27.3.4 getTemplate()

```
size_t Relation::getTemplate ( ) const
```

Get the template of the relation.

#### Returns

Id of the template.

### 14.27.3.5 getTheOtherPerson()

```
size_t Relation::getTheOtherPerson (
    size_t current )
```

Get the other person from relation.

#### Parameters

|                |                 |
|----------------|-----------------|
| <i>current</i> | Current person. |
|----------------|-----------------|

#### Returns

Other person in the relation from current.

### 14.27.3.6 getTheOtherPersonName()

```
const std::string & Relation::getTheOtherPersonName (
    size_t current )
```

Get the name of the other person with respect to the given person as a string.

#### Parameters

|                |                  |
|----------------|------------------|
| <i>current</i> | Given person id. |
|----------------|------------------|

**Returns**

The name of the other person.

**14.27.3.7 getTitle()**

```
const std::string & Relation::getTitle ( ) const
```

Get the title of the relation.

**Returns**

Constant reference to the title.

**14.27.3.8 getTrait()**

```
Trait Relation::getTrait ( )
```

Get the trait of the relation.

**Returns**

Trait of this relation.

**14.27.3.9 isOlderGeneration()**

```
bool Relation::isOlderGeneration (
    size_t id )
```

If the given person from the relation is from older generation.

**Parameters**

|           |                         |
|-----------|-------------------------|
| <i>id</i> | Id of the given person. |
|-----------|-------------------------|

**Returns**

True if the generation difference from given person - other person is  $< 0$ .

**14.27.3.10 isParent()**

```
bool Relation::isParent (
    size_t personId )
```

If the relation is of motherhood or fatherhood trait and the other person is younger.



## Parameters

|                       |  |
|-----------------------|--|
| <i>person↔<br/>Id</i> | Id of the person whether it is father or mother. |
|-----------------------|--|

## Returns

True if all given properties hold, otherwise false.

**14.27.3.11 isSameGeneration()**

```
bool Relation::isSameGeneration ( )
```

If the generation difference is 0.

## Returns

True if the generation difference is 0.

**14.27.3.12 isYoungerGeneration()**

```
bool Relation::isYoungerGeneration (
    size_t id )
```

If the given person from the relation is from younger generation.

## Parameters

|           |                         |
|-----------|-------------------------|
| <i>id</i> | Id of the given person. |
|-----------|-------------------------|

## Returns

True if the generation difference from given person - other person is  $> 0$ .

**14.27.3.13 printHtml()**

```
void Relation::printHtml (
    std::ostream & os,
    const std::string & firstPerson,
    const std::string & secondPerson,
    const std::string & promoted )
```

Print relation to an output stream in HTML format.

## Parameters

|                     |   |
|---------------------|---|
| <i>os</i>           | Given output stream,                      |
| <i>firstPerson</i>  | Name of the first person.                 |
| <i>secondPerson</i> | Name of the second person.                |
| <i>promoted</i>     | What is the name of the promoted version. |

#### 14.27.3.14 readJson()

```
void Relation::readJson (
    const Json::Value & value )
```

Read the relation from JSON value.

##### Parameters

|              |                           |
|--------------|---------------------------|
| <i>value</i> | The JSON value from file. |
|--------------|---------------------------|

#### 14.27.3.15 setId()

```
void Relation::setId (
    size_t id )
```

Set the id of this relation.

##### Parameters

|           |         |
|-----------|---------|
| <i>id</i> | New id. |
|-----------|---------|

#### 14.27.3.16 setPersons()

```
void Relation::setPersons (
    size_t person1,
    size_t person2 )
```

Set persons for this relation.

##### Parameters

|                |                          |
|----------------|--------------------------|
| <i>person1</i> | Id of the first person.  |
| <i>person2</i> | Id of the second person. |

#### 14.27.3.17 setTemplate()

```
void Relation::setTemplate (
    size_t templ )
```

Set the relation template.

##### Parameters

|              |                     |
|--------------|---------------------|
| <i>templ</i> | Id of the template. |
|--------------|---------------------|

### 14.27.3.18 str()

```
std::string Relation::str (
    const std::string & firstPerson,
    const std::string & secondPerson ) const
```

Get the string representation of relation.

#### Parameters

|                     |  |
|---------------------|--|
| <i>firstPerson</i>  | String representing the first person.  |
| <i>secondPerson</i> | String representing the second person. |

#### Returns

String representing the relation.

## 14.27.4 Member Data Documentation

### 14.27.4.1 id\_

```
size_t Relation::id_ [private]
```

Id of this relation.

### 14.27.4.2 person1\_

```
size_t Relation::person1_ [private]
```

First person of the relation.

### 14.27.4.3 person2\_

```
size_t Relation::person2_ [private]
```

Second person of the relation.

### 14.27.4.4 settings\_

```
Settings* Relation::settings_ [private]
```

Used settings holding the templates.

#### 14.27.4.5 template\_

```
size_t Relation::template_ [private]
```

[Relation](#) template it is using.

The documentation for this class was generated from the following files:

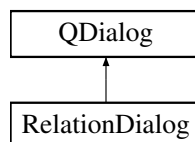
- [core/family\\_tree\\_items.h](#)
- [core/family\\_tree\\_items.cpp](#)

## 14.28 RelationDialog Class Reference

Dialog showing single relation for editing it.

```
#include <dialogs.h>
```

Inheritance diagram for RelationDialog:



### Public Slots

- void [chooseFirstPerson](#) ()  
*Slot for opening dialog to save first person.*
- void [chooseSecondPerson](#) ()  
*Slot for opening dialog to save second person.*
- void [savePerson](#) ([Person](#) \*person)  
*Save person upon choosing it from another dialog.*
- void [saveRelation](#) ()  
*Save relation to the family tree or add new one and save it.*
- void [showTemplateDetails](#) ()  
*Show details about current template.*
- void [switchPersons](#) ()  
*Switch the persons.*

### Public Member Functions

- [RelationDialog](#) ([FamilyTree](#) \*FT, QWidget \*parent=nullptr, [Relation](#) \*rel=nullptr)
- [~RelationDialog](#) ()  
*Default destructor.*

## Private Attributes

- [FamilyTree](#) \* [FT\\_](#)  
*Pointer to the family tree.*
- [Person](#) \* [person1\\_](#)  
*First person of the relation.*
- [Person](#) \* [person2\\_](#)  
*Second person of the relation.*
- [Relation](#) \* [rel\\_](#)  
*Pointer to the edited relation, if adding new one it is a null pointer.*
- bool [saveFirstPerson\\_](#)  
*Whether the first person is to be saved or the second.*
- [Ui::RelationDialog](#) \* [ui](#)  
*User interface of the Qt framework.*

## 14.28.1 Detailed Description

Dialog showing single relation for editing it.

## 14.28.2 Constructor & Destructor Documentation

### 14.28.2.1 RelationDialog()

```
RelationDialog::RelationDialog (
    FamilyTree * FT,
    QWidget * parent = nullptr,
    Relation * rel = nullptr ) [explicit]
```

Default constructor.

#### Parameters

|               |  |
|---------------|--|
| <i>FT</i>     | Pointer to the family tree.              |
| <i>parent</i> | The Qt Widget parent.                    |
| <i>rel</i>    | Which relation is to be added or edited. |

### 14.28.2.2 ~RelationDialog()

```
RelationDialog::~RelationDialog ( )
```

Default destructor.

## 14.28.3 Member Function Documentation

### 14.28.3.1 chooseFirstPerson

```
void RelationDialog::chooseFirstPerson ( ) [slot]
```

Slot for opening dialog to save first person.

#### 14.28.3.2 chooseSecondPerson

```
void RelationDialog::chooseSecondPerson ( ) [slot]
```

Slot for opening dialog to save second person.

#### 14.28.3.3 savePerson

```
void RelationDialog::savePerson (
    Person * person ) [slot]
```

Save person upon choosing it from another dialog.

#### 14.28.3.4 saveRelation

```
void RelationDialog::saveRelation ( ) [slot]
```

Save relation to the family tree or add new one and save it.

#### 14.28.3.5 showTemplateDetails

```
void RelationDialog::showTemplateDetails ( ) [slot]
```

Show details about current template.

#### 14.28.3.6 switchPersons

```
void RelationDialog::switchPersons ( ) [slot]
```

Switch the persons.

### 14.28.4 Member Data Documentation

#### 14.28.4.1 FT\_

```
FamilyTree* RelationDialog::FT_ [private]
```

Pointer to the family tree.

#### 14.28.4.2 person1\_

```
Person* RelationDialog::person1_ [private]
```

First person of the relation.

**14.28.4.3 person2\_**

```
Person* RelationDialog::person2_ [private]
```

Second person of the relation.

**14.28.4.4 rel\_**

```
Relation* RelationDialog::rel_ [private]
```

Pointer to the edited relation, if adding new one it is a null pointer.

**14.28.4.5 saveFirstPerson\_**

```
bool RelationDialog::saveFirstPerson_ [private]
```

Whether the first person is to be saved or the second.

**14.28.4.6 ui**

```
Ui::RelationDialog* RelationDialog::ui [private]
```

User interface of the Qt framework.

The documentation for this class was generated from the following files:

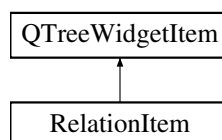
- [graphics/dialogs.h](#)
- [graphics/dialogs.cpp](#)

**14.29 RelationItem Class Reference**

Class for showing relation suggestion and holding the data.

```
#include <graphics_items.h>
```

Inheritance diagram for RelationItem:

**Public Member Functions**

- [RelationItem](#) ([FamilyTree](#) \*FT, const [RelationSuggestion](#) &suggestion, [RelationTemplate](#) \*templ)
- [Relation](#) \* [getRelation](#) ()
- void [refreshTexts](#) ()  
*Update the texts.*

## Private Attributes

- [FamilyTree](#) \* [FT\\_](#)  
*Pointer to the family tree.*
- [Relation](#) [relation\\_](#)  
*Which relation it represents.*

## 14.29.1 Detailed Description

Class for showing relation suggestion and holding the data.

## 14.29.2 Constructor & Destructor Documentation

### 14.29.2.1 RelationItem()

```
RelationItem::RelationItem (
    FamilyTree * FT,
    const RelationSuggestion & suggestion,
    RelationTemplate * templ )
```

Default constructor.

#### Parameters

|                   |  |
|-------------------|--|
| <i>FT</i>         | Pointer to the family tree.            |
| <i>suggestion</i> | Which suggestion this item represents. |
| <i>templ</i>      | Pointer to the used template.          |

## 14.29.3 Member Function Documentation

### 14.29.3.1 getRelation()

```
Relation * RelationItem::getRelation ( )
```

Get the pointer to the relation it represents.

#### Returns

Pointer to the relation.

### 14.29.3.2 refreshTexts()

```
void RelationItem::refreshTexts ( )
```

Update the texts.



## 14.29.4 Member Data Documentation

### 14.29.4.1 FT\_

`FamilyTree* RelationItem::FT_ [private]`

Pointer to the family tree.

### 14.29.4.2 relation\_

`Relation RelationItem::relation_ [private]`

Which relation it represents.

The documentation for this class was generated from the following files:

- `graphics/graphics_items.h`
- `graphics/graphics_items.cpp`

## 14.30 RelationSuggestion Struct Reference

Struct for relation suggestions.

```
#include <family_tree_items.h>
```

### Public Member Functions

- `RelationSuggestion` (size\_t lockedP, size\_t suggestedP, bool par, bool sym, Trait t)

### Public Attributes

- size\_t `lockedPerson`  
*The one locked person.*
- bool `parent`  
*If the locked person should be parent or not.*
- size\_t `suggestedPerson`  
*Suggested person for the relation.*
- bool `symetrical`  
*If the relation is symmetrical or not.*
- Trait `trait`  
*What relation should it be.*

### 14.30.1 Detailed Description

Struct for relation suggestions.

## 14.30.2 Constructor & Destructor Documentation

### 14.30.2.1 RelationSuggestion()

```
RelationSuggestion::RelationSuggestion (
    size_t lockedP,
    size_t suggestedP,
    bool par,
    bool sym,
    Trait t )
```

Default constructor.

## Parameters

|                   |                       |
|-------------------|-----------------------|
| <i>lockedP</i>    | Locked person.        |
| <i>suggestedP</i> | Suggested person.     |
| <i>par</i>        | If it is parent.      |
| <i>sym</i>        | If it is symmetrical. |
| <i>t</i>          | Given trait.          |

### 14.30.3 Member Data Documentation

#### 14.30.3.1 lockedPerson

```
size_t RelationSuggestion::lockedPerson
```

The one locked person.

#### 14.30.3.2 parent

```
bool RelationSuggestion::parent
```

If the locked person should be parent or not.

#### 14.30.3.3 suggestedPerson

```
size_t RelationSuggestion::suggestedPerson
```

Suggested person for the relation.

#### 14.30.3.4 symmetrical

```
bool RelationSuggestion::symmetrical
```

If the relation is symmetrical or not.

#### 14.30.3.5 trait

```
Trait RelationSuggestion::trait
```

What relation should it be.

The documentation for this struct was generated from the following files:

- [core/family\\_tree\\_items.h](#)
- [core/family\\_tree\\_items.cpp](#)

## 14.31 RelationTemplate Class Reference

Template for relation. What is the name, description. What is the generation difference. And how are persons in this relation called.

```
#include <config.h>
```

### Public Member Functions

- [RelationTemplate](#) ()  
*Default constructor.*
- const std::string & [getDescription](#) () const
- const std::string & [getFirstName](#) () const
- int [getGenerationDifference](#) () const
- size\_t [getId](#) () const
- const std::string & [getSecondName](#) () const
- const std::string & [getTitle](#) () const
- [Trait](#) [getTrait](#) () const
- void [readJson](#) (const Json::Value &value)
- std::string & [setDescription](#) ()
- std::string & [setFirstName](#) ()
- void [setGenerationDifference](#) (int diff)
- void [setId](#) (size\_t id)
- std::string & [setSecondName](#) ()
- std::string & [setTitle](#) ()
- void [setTrait](#) ([Trait](#) trait)
- std::string [str](#) () const

### Private Attributes

- std::string [description\\_](#)  
*Description of the relation.*
- std::string [firstName\\_](#)  
*First name in the relation.*
- int [generationDiff\\_](#)  
*Generation difference: first generation - second generation.*
- size\_t [id\\_](#)  
*Special ID of this template.*
- std::string [secondName\\_](#)  
*Second name in the relation.*
- std::string [title\\_](#)  
*Title of the relation.*
- [Trait](#) [trait\\_](#)  
*Special trait of this relation.*

### 14.31.1 Detailed Description

Template for relation. What is the name, description. What is the generation difference. And how are persons in this relation called.

## 14.31.2 Constructor & Destructor Documentation

### 14.31.2.1 RelationTemplate()

```
RelationTemplate::RelationTemplate ( )
```

Default constructor.

## 14.31.3 Member Function Documentation

### 14.31.3.1 getDescription()

```
const std::string & RelationTemplate::getDescription ( ) const
```

Get the description.

#### Returns

Constant reference to the description.

### 14.31.3.2 getFirstName()

```
const std::string & RelationTemplate::getFirstName ( ) const
```

Get the first name of the relation.

#### Returns

Constant reference to the first name.

### 14.31.3.3 getGenerationDifference()

```
int RelationTemplate::getGenerationDifference ( ) const
```

Get the generation difference.

#### Returns

The difference between first and second generation.

### 14.31.3.4 getId()

```
size_t RelationTemplate::getId ( ) const
```

Get the ID of this relation template.

#### Returns

ID of this relation template.

#### 14.31.3.5 getSecondName()

```
const std::string & RelationTemplate::getSecondName ( ) const
```

Get the second name of the relation.

##### Returns

Constant reference to the second name.

#### 14.31.3.6 getTitle()

```
const std::string & RelationTemplate::getTitle ( ) const
```

Get the title of the relation.

##### Returns

Constant reference to the title.

#### 14.31.3.7 getTrait()

```
Trait RelationTemplate::getTrait ( ) const
```

Get the trait of the relationship.

##### Returns

Trait of the relationship.

#### 14.31.3.8 readJson()

```
void RelationTemplate::readJson (
    const Json::Value & value )
```

Read data from JSON.

##### Parameters

|              |                   |
|--------------|-------------------|
| <i>value</i> | Given JSON value. |
|--------------|-------------------|

#### 14.31.3.9 setDescription()

```
std::string & RelationTemplate::setDescription ( )
```

Set the description of the relation by giving the reference to the description.

**Returns**

Reference to the description.

**14.31.3.10 setFirstName()**

```
std::string & RelationTemplate::setFirstName ( )
```

Set the first name by giving the reference to the first name.

**Returns**

Reference to the first name.

**14.31.3.11 setGenerationDifference()**

```
void RelationTemplate::setGenerationDifference (
    int diff )
```

Set the generation difference (first - second).

**Parameters**

|             |                   |
|-------------|-------------------|
| <i>diff</i> | Given difference. |
|-------------|-------------------|

**14.31.3.12 setId()**

```
void RelationTemplate::setId (
    size_t id )
```

Set the ID of this relation template.

**Parameters**

|           |         |
|-----------|---------|
| <i>id</i> | New ID. |
|-----------|---------|

**14.31.3.13 setSecondName()**

```
std::string & RelationTemplate::setSecondName ( )
```

Set the second name by giving the reference to the second name.

**Returns**

Reference to the second name.

#### 14.31.3.14 setTitle()

```
std::string & RelationTemplate::setTitle ( )
```

Set the title of relation by giving the reference to the title.

##### Returns

Reference to the title.

#### 14.31.3.15 setTrait()

```
void RelationTemplate::setTrait (
    Trait trait )
```

Set the trait for the relation.

##### Parameters

|              |              |
|--------------|--------------|
| <i>trait</i> | Given trait. |
|--------------|--------------|

#### 14.31.3.16 str()

```
std::string RelationTemplate::str ( ) const
```

Represent relation template as a string.

##### Returns

Newly constructed string representing relation template.

### 14.31.4 Member Data Documentation

#### 14.31.4.1 description\_

```
std::string RelationTemplate::description_ [private]
```

Description of the relation.

#### 14.31.4.2 firstName\_

```
std::string RelationTemplate::firstName_ [private]
```

First name in the relation.



#### 14.31.4.3 generationDiff\_

```
int RelationTemplate::generationDiff_ [private]
```

Generation difference: first generation - second generation.

#### 14.31.4.4 id\_

```
size_t RelationTemplate::id_ [private]
```

Special ID of this template.

#### 14.31.4.5 secondName\_

```
std::string RelationTemplate::secondName_ [private]
```

Second name in the relation.

#### 14.31.4.6 title\_

```
std::string RelationTemplate::title_ [private]
```

Title of the relation.

#### 14.31.4.7 trait\_

```
Trait RelationTemplate::trait_ [private]
```

Special trait of this relation.

The documentation for this class was generated from the following files:

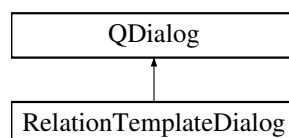
- [core/config.h](#)
- [core/config.cpp](#)

## 14.32 RelationTemplateDialog Class Reference

Dialog showing all templates and possibly editing them or creating new ones.

```
#include <dialogs.h>
```

Inheritance diagram for RelationTemplateDialog:



## Public Slots

- void [addDefaults](#) ()  
*Add default templates.*
- void [addNewTemplate](#) ()  
*Clear boxes and set them for new template.*
- void [removeTemplate](#) ()  
*Remove current template from the templates.*
- void [saveTemplate](#) ()  
*Either save edits to the template or add new item.*
- void [showTemplate](#) (QTreeWidgetItem \*item)

## Public Member Functions

- [RelationTemplateDialog](#) (FamilyTree \*FT, QWidget \*parent=nullptr)
- [~RelationTemplateDialog](#) ()  
*Default destructor.*

## Private Member Functions

- void [clear](#) ()  
*Clear all text boxes.*
- void [refreshListWidget](#) ()  
*Refresh list of all relation templates.*
- void [setEnabledWidgets](#) (bool enabled)
- void [saveTemplate](#) (RelationTemplate \*rel)

## Private Attributes

- bool [addingNew\\_](#)  
*When we are adding new template.*
- FamilyTree \* [FT\\_](#)  
*Pointer to the family tree.*
- Ui::RelationTemplateDialog \* [ui](#)  
*User interface of the Qt framework.*

### 14.32.1 Detailed Description

Dialog showing all templates and possibly editing them or creating new ones.

### 14.32.2 Constructor & Destructor Documentation

#### 14.32.2.1 RelationTemplateDialog()

```
RelationTemplateDialog::RelationTemplateDialog (
    FamilyTree * FT,
    QWidget * parent = nullptr ) [explicit]
```

Default constructor.

## Parameters

|               |                             |
|---------------|-----------------------------|
| <i>FT</i>     | Pointer to the family tree. |
| <i>parent</i> | The Qt Widget parent.       |

**14.32.2.2 ~RelationTemplateDialog()**

```
RelationTemplateDialog::~RelationTemplateDialog ( )
```

Default destructor.

**14.32.3 Member Function Documentation****14.32.3.1 addDefaults**

```
void RelationTemplateDialog::addDefaults ( ) [slot]
```

Add default templates.

**14.32.3.2 addNewTemplate**

```
void RelationTemplateDialog::addNewTemplate ( ) [slot]
```

Clear boxes and set them for new template.

**14.32.3.3 clear()**

```
void RelationTemplateDialog::clear ( ) [private]
```

Clear all text boxes.

**14.32.3.4 refreshListWidget()**

```
void RelationTemplateDialog::refreshListWidget ( ) [private]
```

Refresh list of all relation templates.

**14.32.3.5 removeTemplate**

```
void RelationTemplateDialog::removeTemplate ( ) [slot]
```

Remove current template from the templates.

**14.32.3.6 saveTemplate [1/2]**

```
void RelationTemplateDialog::saveTemplate ( ) [slot]
```

Either save edits to the template or add new item.

**14.32.3.7 saveTemplate() [2/2]**

```
void RelationTemplateDialog::saveTemplate (
    RelationTemplate * rel ) [private]
```

Store the given template from ui boxes.

## Parameters

|            |   |
|------------|---|
| <i>rel</i> | Pointer to which relation to overwrite. |
|------------|---|

**14.32.3.8 setEnabledWidgets()**

```
void RelationTemplateDialog::setEnabledWidgets (
    bool enabled ) [private]
```

Set enabled for all widgets.

## Parameters

|                |  |
|----------------|--|
| <i>enabled</i> | If the widgets should be enabled or not. |
|----------------|--|

**14.32.3.9 showTemplate**

```
void RelationTemplateDialog::showTemplate (
    QTreeWidgetItem * item ) [slot]
```

Show template details to its text boxes.

## Parameters

|             |                         |
|-------------|-------------------------|
| <i>item</i> | Which item was clicked. |
|-------------|-------------------------|

**14.32.4 Member Data Documentation****14.32.4.1 addingNew\_**

```
bool RelationTemplateDialog::addingNew_ [private]
```

When we are adding new template.

**14.32.4.2 FT\_**

```
FamilyTree* RelationTemplateDialog::FT_ [private]
```

Pointer to the family tree.

**14.32.4.3 ui**

```
Ui::RelationTemplateDialog* RelationTemplateDialog::ui [private]
```

User interface of the Qt framework.

The documentation for this class was generated from the following files:

- [graphics/dialogs.h](#)
- [graphics/dialogs.cpp](#)

## 14.33 Settings Class Reference

Class holding all data about the settings of the app and all templates.

```
#include <config.h>
```

### Public Member Functions

- [Settings](#) ()  
*Default constructor.*
- void [addDefaultEvents](#) ()  
*Add default event templates.*
- void [addDefaultRelations](#) ()  
*Add default relation templates.*
- size\_t [addEventTemplate](#) ([EventTemplate](#) &et)
- size\_t [addRelationTemplate](#) ([RelationTemplate](#) &rt)
- void [addTag](#) (const std::string &tag)
- void [clear](#) ()  
*Clear all data.*
- void [exportTemplates](#) (std::ostream &os, const std::vector< size\_t > &relIds, const std::vector< size\_t > &eventIds)
- const [AppSettings](#) & [getAppSettings](#) () const
- std::optional< [EventTemplate](#) \* > [getEventTemplate](#) (size\_t id) const
- const std::map< size\_t, std::unique\_ptr< [EventTemplate](#) > > & [getEventTemplates](#) () const
- size\_t [getGlobalMainPerson](#) () const
- std::optional< [RelationTemplate](#) \* > [getRelationTemplate](#) (size\_t id) const
- const std::map< size\_t, std::unique\_ptr< [RelationTemplate](#) > > & [getRelationTemplates](#) () const
- const std::vector< std::string > & [getTags](#) () const
- size\_t [importTemplates](#) (const Json::Value &value)
- std::pair< size\_t, size\_t > [indexes](#) ()
- void [readJson](#) (const Json::Value &value)
- void [reIndex](#) (size\_t plusEvent, size\_t plusRel)
- void [removeEventTemplate](#) (size\_t index)
- void [removeRelationTemplate](#) (size\_t index)
- void [removeTag](#) (const std::string &tag)
- [AppSettings](#) & [setAppSettings](#) ()
- void [setGlobalMainPerson](#) (size\_t id)

### Private Member Functions

- void [addDefaultEvent](#) (const std::string &title, const std::string &description, bool date, bool place, bool morePeople, std::vector< std::string > roles)
- void [addDefaultRelation](#) (const std::string &title, const std::string &description, const std::string &firstName, const std::string &secondName, int genDiff, [Trait](#) trait)

## Private Attributes

- [AppSettings appSettings\\_](#)  
*Application settings.*
- [size\\_t eventTemplateIndex\\_](#)  
*First empty index for event template.*
- [std::map< size\\_t, std::unique\\_ptr< EventTemplate > > eventTemplates\\_](#)  
*All events templates.*
- [size\\_t globalMainPerson\\_](#)  
*Main person that is global, so that on app start it will be chosen as the main person.*
- [size\\_t relationTemplateIndex\\_](#)  
*First empty index for relation template.*
- [std::map< size\\_t, std::unique\\_ptr< RelationTemplate > > relationsTemplates\\_](#)  
*All relationsTemplates.*
- [std::vector< std::string > tags\\_](#)  
*All tags.*

## 14.33.1 Detailed Description

Class holding all data about the settings of the app and all templates.

## 14.33.2 Constructor & Destructor Documentation

### 14.33.2.1 Settings()

```
Settings::Settings ( )
```

Default constructor.

## 14.33.3 Member Function Documentation

### 14.33.3.1 addDefaultEvent()

```
void Settings::addDefaultEvent (
    const std::string & title,
    const std::string & description,
    bool date,
    bool place,
    bool morePeople,
    std::vector< std::string > roles ) [private]
```

Add single default event.

#### Parameters

|                    |  |
|--------------------|--|
| <i>title</i>       | Title of the new event template.             |
| <i>description</i> | Description of the new event template.       |
| <i>date</i>        | If the event template contains date.         |
| <i>place</i>       | If the event template contains date.         |
| <i>morePeople</i>  | If more people are present.                  |
| <i>roles</i>       | Which roles, if any, the event template has. |

**14.33.3.2 addDefaultEvents()**

```
void Settings::addDefaultEvents ( )
```

Add default event templates.

**14.33.3.3 addDefaultRelation()**

```
void Settings::addDefaultRelation (
    const std::string & title,
    const std::string & description,
    const std::string & firstName,
    const std::string & secondName,
    int genDiff,
    Trait trait ) [private]
```

Add single default relation.

**Parameters**

|                    |   |
|--------------------|---|
| <i>title</i>       | Title of the new relation template.       |
| <i>description</i> | Description of the new relation template. |
| <i>firstName</i>   | First name of the person.                 |
| <i>secondName</i>  | Second name of the person.                |
| <i>genDiff</i>     | Generation difference.                    |
| <i>trait</i>       | Which trait it has.                       |

**14.33.3.4 addDefaultRelations()**

```
void Settings::addDefaultRelations ( )
```

Add default relation templates.

**14.33.3.5 addEventTemplate()**

```
size_t Settings::addEventTemplate (
    EventTemplate & et )
```

Add event template to the vector. Also re-index if necessary (id = 0).

**Parameters**

|           |                       |
|-----------|-----------------------|
| <i>et</i> | Given event template. |
|-----------|-----------------------|

**Returns**

Id of the template.

**14.33.3.6 addRelationTemplate()**

```
size_t Settings::addRelationTemplate (
    RelationTemplate & rt )
```

Add relation event to the vector. Also re-index if necessary (id = 0).

**Parameters**

|           |                          |
|-----------|--------------------------|
| <i>rt</i> | Given relation template. |
|-----------|--------------------------|

**Returns**

Id of the template.

**14.33.3.7 addTag()**

```
void Settings::addTag (
    const std::string & tag )
```

Add tag to the vector. If it is not present.

**Parameters**

|            |          |
|------------|----------|
| <i>tag</i> | New tag. |
|------------|----------|

**14.33.3.8 clear()**

```
void Settings::clear ( )
```

Clear all data.

**14.33.3.9 exportTemplates()**

```
void Settings::exportTemplates (
    std::ostream & os,
    const std::vector< size_t > & relIds,
    const std::vector< size_t > & eventIds )
```

Export only certain templates in JSON.

**Parameters**

|                 |  |
|-----------------|--|
| <i>os</i>       | Which output stream to use for writing the data. |
| <i>relIds</i>   | Vector of relation templates ids.                |
| <i>eventIds</i> | Vector of event templates ids.                   |



#### 14.33.3.10 getAppSettings()

```
const AppSettings & Settings::getAppSettings ( ) const
```

Get the app settings.

##### Returns

Constant reference to the app settings.

#### 14.33.3.11 getEventTemplate()

```
std::optional< EventTemplate * > Settings::getEventTemplate (
    size_t id ) const
```

Get exactly one event template.

##### Parameters

|           |                      |
|-----------|----------------------|
| <i>id</i> | The id of the event. |
|-----------|----------------------|

##### Returns

Optionally pointer to the template or nothing.

#### 14.33.3.12 getEventTemplates()

```
const std::map< size_t, std::unique_ptr< EventTemplate > > & Settings::getEventTemplates ( )
const
```

Get all the event templates.

##### Returns

Constant reference to the vector of all templates.

#### 14.33.3.13 getGlobalMainPerson()

```
size_t Settings::getGlobalMainPerson ( ) const
```

Get the global main person.

##### Returns

Id of the main person.

#### 14.33.3.14 getRelationTemplate()

```
std::optional< RelationTemplate * > Settings::getRelationTemplate (
    size_t id ) const
```

Get exactly one relation template.

**Parameters**

|           |                         |
|-----------|-------------------------|
| <i>id</i> | The id of the relation. |
|-----------|-------------------------|

**Returns**

Optionally pointer to the template or nothing.

**14.33.3.15 getRelationTemplates()**

```
const std::map< size_t, std::unique_ptr< RelationTemplate > > & Settings::getRelationTemplates  
( ) const
```

Get all the relation templates.

**Returns**

Constant reference to the vector of all templates.

**14.33.3.16 getTags()**

```
const std::vector< std::string > & Settings::getTags ( ) const
```

Return the reference to all tags.

**Returns**

Constant reference to the container.

**14.33.3.17 importTemplates()**

```
size_t Settings::importTemplates (   
    const Json::Value & value )
```

Import templates from JSON value.

**Parameters**

|              |                   |
|--------------|-------------------|
| <i>value</i> | Given JSON Value. |
|--------------|-------------------|

**Returns**

Number of imported templates.

**14.33.3.18 indexes()**

```
std::pair< size_t, size_t > Settings::indexes ( )
```

Get the free index for templates.

#### Returns

Pair of event templates index and then relation templates.

#### 14.33.3.19 readJson()

```
void Settings::readJson (
    const Json::Value & value )
```

Read the configuration from its JSON value.

#### Parameters

|              |                   |
|--------------|-------------------|
| <i>value</i> | Given JSON value. |
|--------------|-------------------|

#### 14.33.3.20 reIndex()

```
void Settings::reIndex (
    size_t plusEvent,
    size_t plusRel )
```

Re-index all templates to plus given number.

#### Parameters

|                  |  |
|------------------|--|
| <i>plusEvent</i> | What is the number to add to event templates.    |
| <i>plusRel</i>   | What is the number to add to relation templates. |

#### 14.33.3.21 removeEventTemplate()

```
void Settings::removeEventTemplate (
    size_t index )
```

Remove event template from all events.

#### Parameters

|              |   |
|--------------|---|
| <i>index</i> | On which position the event in vector is. |
|--------------|---|

#### 14.33.3.22 removeRelationTemplate()

```
void Settings::removeRelationTemplate (
    size_t index )
```

Remove relation template from all relations.

## Parameters

|              |  |
|--------------|--|
| <i>index</i> | On which position the relation in vector is. |
|--------------|--|

**14.33.3.23 removeTag()**

```
void Settings::removeTag (
    const std::string & tag )
```

Remove tag from the container.

## Parameters

|            |                               |
|------------|-------------------------------|
| <i>tag</i> | Delete same tag if it exists. |
|------------|-------------------------------|

**14.33.3.24 setAppSettings()**

```
AppSettings & Settings::setAppSettings ( )
```

Get the reference to the app settings to edit it.

## Returns

Reference to the app settings.

**14.33.3.25 setGlobalMainPerson()**

```
void Settings::setGlobalMainPerson (
    size_t id )
```

Set the global main person.

## Parameters

|           |                            |
|-----------|----------------------------|
| <i>id</i> | Id of the new main person. |
|-----------|----------------------------|

**14.33.4 Member Data Documentation****14.33.4.1 appSettings\_**

```
AppSettings Settings::appSettings_ [private]
```

Application settings.

#### 14.33.4.2 eventTemplateIndex\_

```
size_t Settings::eventTemplateIndex_ [private]
```

First empty index for event template.

#### 14.33.4.3 eventTemplates\_

```
std::map<size_t, std::unique_ptr<EventTemplate>> > Settings::eventTemplates_ [private]
```

All events templates.

#### 14.33.4.4 globalMainPerson\_

```
size_t Settings::globalMainPerson_ [private]
```

Main person that is global, so that on app start it will be chosen as the main person.

#### 14.33.4.5 relationsTemplates\_

```
std::map<size_t, std::unique_ptr<RelationTemplate>> > Settings::relationsTemplates_ [private]
```

All relationsTemplates.

#### 14.33.4.6 relationTemplateIndex\_

```
size_t Settings::relationTemplateIndex_ [private]
```

First empty index for relation template.

#### 14.33.4.7 tags\_

```
std::vector<std::string> Settings::tags_ [private]
```

All tags.

The documentation for this class was generated from the following files:

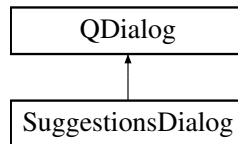
- core/[config.h](#)
- core/[config.cpp](#)

## 14.34 SuggestionsDialog Class Reference

Dialog for relation suggestions.

```
#include <dialogs.h>
```

Inheritance diagram for SuggestionsDialog:



### Public Slots

- void [removeSuggestion](#) ()  
*Remove current suggestion.*
- void [save](#) ()  
*Save all suggestions.*
- void [selectedSuggestion](#) (QTreeWidgetItem \*item)
- void [swapPersons](#) ()  
*Swap persons roles.*
- void [switchedTemplate](#) ()  
*Switch the template description.*

### Public Member Functions

- [SuggestionsDialog](#) (FamilyTree \*FT, std::vector< [RelationSuggestion](#) > suggestions, QWidget \*parent=nullptr)
- [~SuggestionsDialog](#) ()  
*Default destructor.*

### Private Attributes

- std::vector< [RelationTemplate](#) \* > [fatherhoods](#)  
*Vector of all templates for fatherhood relations.*
- [FamilyTree](#) \* [FT\\_](#)  
*Pointer to the family tree.*
- std::vector< [RelationTemplate](#) \* > [motherhoods](#)  
*Vector of all templates for motherhood relations.*
- std::vector< [RelationTemplate](#) \* > [partnerships](#)  
*Vector of all templates for partnership relations.*
- std::vector< [RelationTemplate](#) \* > [siblings](#)  
*Vector of all templates for siblings relations.*
- Ui::SuggestionsDialog \* [ui](#)  
*Used Qt user interface.*

### 14.34.1 Detailed Description

Dialog for relation suggestions.

### 14.34.2 Constructor & Destructor Documentation

#### 14.34.2.1 SuggestionsDialog()

```
SuggestionsDialog::SuggestionsDialog (
    FamilyTree * FT,
    std::vector< RelationSuggestion > suggestions,
    QWidget * parent = nullptr )
```

Default constructor.

##### Parameters

|                    |                                    |
|--------------------|------------------------------------|
| <i>FT</i>          | Pointer to the family tree.        |
| <i>suggestions</i> | All the suggestions for relations. |
| <i>parent</i>      | The Qt Widget parent.              |

#### 14.34.2.2 ~SuggestionsDialog()

```
SuggestionsDialog::~SuggestionsDialog ( )
```

Default destructor.

### 14.34.3 Member Function Documentation

#### 14.34.3.1 removeSuggestion

```
void SuggestionsDialog::removeSuggestion ( ) [slot]
```

Remove current suggestion.

#### 14.34.3.2 save

```
void SuggestionsDialog::save ( ) [slot]
```

Save all suggestions.

#### 14.34.3.3 selectedSuggestion

```
void SuggestionsDialog::selectedSuggestion (
    QTreeWidgetItem * item ) [slot]
```

When a suggestion is clicked.



## Parameters

|             |                               |
|-------------|-------------------------------|
| <i>item</i> | Which suggestion was clicked. |
|-------------|-------------------------------|

#### 14.34.3.4 swapPersons

```
void SuggestionsDialog::swapPersons ( ) [slot]
```

Swap persons roles.

#### 14.34.3.5 switchedTemplate

```
void SuggestionsDialog::switchedTemplate ( ) [slot]
```

Switch the template description.

### 14.34.4 Member Data Documentation

#### 14.34.4.1 fatherhoods

```
std::vector<RelationTemplate*> SuggestionsDialog::fatherhoods [private]
```

Vector of all templates for fatherhood relations.

#### 14.34.4.2 FT\_

```
FamilyTree* SuggestionsDialog::FT_ [private]
```

Pointer to the family tree.

#### 14.34.4.3 motherhoods

```
std::vector<RelationTemplate*> SuggestionsDialog::motherhoods [private]
```

Vector of all templates for motherhood relations.

#### 14.34.4.4 partnerships

```
std::vector<RelationTemplate*> SuggestionsDialog::partnerships [private]
```

Vector of all templates for partnership relations.

#### 14.34.4.5 siblings

```
std::vector<RelationTemplate*> SuggestionsDialog::siblings [private]
```

Vector of all templates for siblings relations.

#### 14.34.4.6 ui

```
Ui::SuggestionsDialog* SuggestionsDialog::ui [private]
```

Used Qt user interface.

The documentation for this class was generated from the following files:

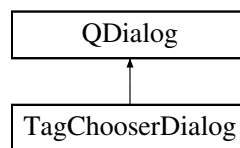
- [graphics/dialogs.h](#)
- [graphics/dialogs.cpp](#)

### 14.35 TagChooserDialog Class Reference

Dialog for editing or creating tags for a person.

```
#include <dialogs.h>
```

Inheritance diagram for TagChooserDialog:



#### Public Slots

- void [changeVisibleWidgets](#) (int index)
- void [saveTag](#) ()  
*Save the tag.*

#### Public Member Functions

- [TagChooserDialog](#) ([FamilyTree](#) \*FT, const QString &tag, const QString &value, QWidget \*parent=nullptr)
- [~TagChooserDialog](#) ()  
*Default destructor.*

#### Private Attributes

- [FamilyTree](#) \* FT\_  
*Pointer to the family tree.*
- bool [newOne](#)\_  
*If the tag is being edited or new one is being created.*
- Ui::TagChooserDialog \* ui  
*User interface of the Qt framework.*

## 14.35.1 Detailed Description

Dialog for editing or creating tags for a person.

## 14.35.2 Constructor & Destructor Documentation

### 14.35.2.1 TagChooserDialog()

```
TagChooserDialog::TagChooserDialog (
    FamilyTree * FT,
    const QString & tag,
    const QString & value,
    QWidget * parent = nullptr )
```

Default constructor.

#### Parameters

|               |                                     |
|---------------|-------------------------------------|
| <i>FT</i>     | Pointer to the family tree.         |
| <i>tag</i>    | What to show as a tag.              |
| <i>value</i>  | What to show as a value of the tag. |
| <i>parent</i> | The Qt Widget parent.               |

### 14.35.2.2 ~TagChooserDialog()

```
TagChooserDialog::~TagChooserDialog ( )
```

Default destructor.

## 14.35.3 Member Function Documentation

### 14.35.3.1 changeVisibleWidgets

```
void TagChooserDialog::changeVisibleWidgets (
    int index ) [slot]
```

Slot for showing/hiding custom tag edit.

#### Parameters

|              |  |
|--------------|--|
| <i>index</i> | Which index was from the combo box chosen. |
|--------------|--|

### 14.35.3.2 saveTag

```
void TagChooserDialog::saveTag ( ) [slot]
```

Save the tag.

## 14.35.4 Member Data Documentation

### 14.35.4.1 FT\_

`FamilyTree* TagChooserDialog::FT_ [private]`

Pointer to the family tree.

### 14.35.4.2 newOne\_

`bool TagChooserDialog::newOne_ [private]`

If the tag is being edited or new one is being created.

### 14.35.4.3 ui

`Ui::TagChooserDialog* TagChooserDialog::ui [private]`

User interface of the Qt framework.

The documentation for this class was generated from the following files:

- [graphics/dialogs.h](#)
- [graphics/dialogs.cpp](#)

## 14.36 VirtualDrive Class Reference

Class for virtualization of folder structure for each person.

```
#include <person.h>
```

### Public Member Functions

- [VirtualDrive](#) ()=default  
*Default constructor.*
- [VirtualDrive](#) (const std::string &name)
- void [addFile](#) (size\_t fileId)
- [VirtualDrive](#) \* [addSubdrive](#) ()
- [VirtualDrive](#) \* [addSubdrive](#) (const std::string &name)
- void [copyDrive](#) (const [VirtualDrive](#) &other, size\_t plusIndex=0)
- const std::vector< size\_t > & [getFiles](#) () const
- const std::string & [getName](#) () const
- const std::vector< std::unique\_ptr< [VirtualDrive](#) > > & [getSubdrives](#) () const
- void [readJson](#) (const Json::Value &value)
- void [remove](#) (const std::string &subfolderName)
- void [removeFile](#) (size\_t id)
- void [rename](#) (const std::string &name)
- void [usedFiles](#) (std::vector< size\_t > &used) const

**Private Attributes**

- `std::vector< size_t > files_`  
*Container of all files.*
- `std::string name_`  
*Name of the folder.*
- `std::vector< std::unique_ptr< VirtualDrive > > subdrives_`  
*Container of pointers to the sub-folders.*

**14.36.1 Detailed Description**

Class for virtualization of folder structure for each person.

**14.36.2 Constructor & Destructor Documentation****14.36.2.1 VirtualDrive() [1/2]**

```
VirtualDrive::VirtualDrive ( ) [default]
```

Default constructor.

**14.36.2.2 VirtualDrive() [2/2]**

```
VirtualDrive::VirtualDrive (
    const std::string & name ) [explicit]
```

Constructor for known name.

**Parameters**

|             |                      |
|-------------|----------------------|
| <i>name</i> | Name of this folder. |
|-------------|----------------------|

**14.36.3 Member Function Documentation****14.36.3.1 addFile()**

```
void VirtualDrive::addFile (
    size_t fileId )
```

Add file to the folder by its id. Only if it is not already there.

**Parameters**

|                     |                 |
|---------------------|-----------------|
| <i>file↔<br/>Id</i> | Id of the file. |
|---------------------|-----------------|

#### 14.36.3.2 addSubdrive() [1/2]

```
VirtualDrive * VirtualDrive::addSubdrive ( )
```

Add sub-folder to this folder.

##### Returns

Pointer to the new sub-folder.

#### 14.36.3.3 addSubdrive() [2/2]

```
VirtualDrive * VirtualDrive::addSubdrive (
    const std::string & name )
```

Add sub-folder to this folder.

##### Parameters

|             |                         |
|-------------|-------------------------|
| <i>name</i> | Name of the sub-folder. |
|-------------|-------------------------|

##### Returns

Pointer to the new sub-folder.

#### 14.36.3.4 copyDrive()

```
void VirtualDrive::copyDrive (
    const VirtualDrive & other,
    size_t plusIndex = 0 )
```

Recursively copy the other drive. Used for import mainly.

##### Parameters

|                  |  |
|------------------|--|
| <i>other</i>     | Other drive.                             |
| <i>plusIndex</i> | What should be added to each file index. |

#### 14.36.3.5 getFiles()

```
const std::vector< size_t > & VirtualDrive::getFiles ( ) const
```

Get constant reference to the container of files.

##### Returns

Constant reference to the container of files.

### 14.36.3.6 getName()

```
const std::string & VirtualDrive::getName ( ) const
```

Get the name of the folder.

#### Returns

Name of the folder.

### 14.36.3.7 getSubdrives()

```
const std::vector< std::unique_ptr< VirtualDrive > > & VirtualDrive::getSubdrives ( ) const
```

Get constant reference to the container of files.

#### Returns

Constant reference to the container of pointers to virtual drives.

### 14.36.3.8 readJson()

```
void VirtualDrive::readJson (
    const Json::Value & value )
```

Load data from JSON value.

#### Parameters

|              |                   |
|--------------|-------------------|
| <i>value</i> | Given JSON value. |
|--------------|-------------------|

### 14.36.3.9 remove()

```
void VirtualDrive::remove (
    const std::string & subfolderName )
```

Remove sub-folder and all its sub-folders and files.

#### Parameters

|                      |                         |
|----------------------|-------------------------|
| <i>subfolderName</i> | Name of the sub-folder. |
|----------------------|-------------------------|

### 14.36.3.10 removeFile()

```
void VirtualDrive::removeFile (
    size_t id )
```

Remove one file in this folder.



## Parameters

|           |                 |
|-----------|-----------------|
| <i>id</i> | Id of the file. |
|-----------|-----------------|

**14.36.3.11 rename()**

```
void VirtualDrive::rename (
    const std::string & name )
```

Rename this folder.

## Parameters

|             |                          |
|-------------|--------------------------|
| <i>name</i> | New name of this folder. |
|-------------|--------------------------|

**14.36.3.12 usedFiles()**

```
void VirtualDrive::usedFiles (
    std::vector< size_t > & used ) const
```

Vector of all used files.

## Parameters

|             |   |
|-------------|---|
| <i>used</i> | Vector of all files contained in this folder and sub-folders. |
|-------------|---|

**14.36.4 Member Data Documentation****14.36.4.1 files\_**

```
std::vector<size_t> VirtualDrive::files_ [private]
```

Container of all files.

**14.36.4.2 name\_**

```
std::string VirtualDrive::name_ [private]
```

Name of the folder.

**14.36.4.3 subdrives\_**

```
std::vector<std::unique_ptr<VirtualDrive> > VirtualDrive::subdrives_ [private]
```

Container of pointers to the sub-folders.

The documentation for this class was generated from the following files:

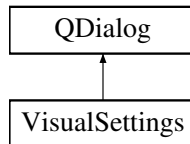
- core/[person.h](#)
- core/[person.cpp](#)

## 14.37 VisualSettings Class Reference

Dialog for editing and showing graphic settings.

```
#include <dialogs.h>
```

Inheritance diagram for VisualSettings:



### Public Slots

- void [highlightedColor](#) ()  
*Show dialog for color change upon clicking on highlighted color change.*
- void [lineColor](#) ()  
*Change the color of line.*
- void [refresh](#) ()  
*Upon clicking the refresh button load the values and show the graphics.*
- void [probandColor](#) ()  
*Show dialog for color change upon clicking on proband color change.*
- void [promotedColor](#) ()  
*Show dialog for color change upon clicking on promoted color change.*
- void [saveAndExit](#) ()  
*Emit signal for saving the configuration and exit the dialog.*
- void [standardColor](#) ()  
*Show dialog for color change upon clicking on standard color change.*
- void [textColor](#) ()  
*Change the color of a text.*

### Signals

- void [saveSettings](#) (QColor [probandColor](#), QColor [promotedColor](#), QColor [standardColor](#), QColor [highlightedColor](#), int sizeX, int sizeY, QFont font, QColor [lineColor](#), QColor [textColor](#), int lineWidth, int rounding)

### Public Member Functions

- [VisualSettings](#) (QColor [probandColor](#), QColor [promotedColor](#), QColor [standardColor](#), QColor highlighted, int sizeX, int sizeY, QFont font, QPen \*pen, QPen \*textPen, int borderRadius, QWidget \*parent=nullptr)
- [~VisualSettings](#) ()  
*Default destructor.*

## Private Attributes

- int `borderRadius_`  
*Border radius of the box.*
- QFont `font_`  
*Used font.*
- QColor `highlightedColor_`  
*Color for highlighting the box.*
- QPen `pen_`  
*Which pen is used for lines.*
- QColor `probandColor_`  
*Color for proband.*
- QColor `promotedColor_`  
*Color for promoted relations.*
- QGraphicsScene \* `scene`  
*Used graphics scene.*
- int `sizeX_`  
*Size of the box in x axis.*
- int `sizeY_`  
*Size of the box in y axis.*
- QColor `standardColor_`  
*Color for standard relations.*
- QPen `textPen_`  
*Pen used for texts.*
- Ui::VisualSettings \* `ui`  
*Used Qt user interface.*

## 14.37.1 Detailed Description

Dialog for editing and showing graphic settings.

## 14.37.2 Constructor & Destructor Documentation

### 14.37.2.1 VisualSettings()

```
VisualSettings::VisualSettings (
    QColor probandColor,
    QColor promotedColor,
    QColor standardColor,
    QColor highlighted,
    int sizeX,
    int sizeY,
    QFont font,
    QPen * pen,
    QPen * textPen,
    int borderRadius,
    QWidget * parent = nullptr )
```

Default constructor.

## Parameters

|                      |                                       |
|----------------------|---------------------------------------|
| <i>probandColor</i>  | Color used for proband.               |
| <i>promotedColor</i> | Color for promoted Relations.         |
| <i>standardColor</i> | Standard color.                       |
| <i>highlighted</i>   | Color for highlighting the box.       |
| <i>sizeX</i>         | Size of the box in x axis.            |
| <i>sizeY</i>         | Size of the box in y axis.            |
| <i>font</i>          | Used font for the text in the box.    |
| <i>pen</i>           | Which pen is used for lines.          |
| <i>textPen</i>       | Which pen is used for text.           |
| <i>borderRadius</i>  | What is the border radius of the box. |
| <i>parent</i>        | The Qt Widget parent.                 |

**14.37.2.2 ~VisualSettings()**

```
VisualSettings::~VisualSettings ( )
```

Default destructor.

**14.37.3 Member Function Documentation****14.37.3.1 highlightedColor**

```
void VisualSettings::highlightedColor ( ) [slot]
```

Show dialog for color change upon clicking on highlighted color change.

**14.37.3.2 lineColor**

```
void VisualSettings::lineColor ( ) [slot]
```

Change the color of line.

**14.37.3.3 probandColor**

```
void VisualSettings::probandColor ( ) [slot]
```

Show dialog for color change upon clicking on proband color change.

**14.37.3.4 promotedColor**

```
void VisualSettings::promotedColor ( ) [slot]
```

Show dialog for color change upon clicking on promoted color change.

#### 14.37.3.5 refresh

```
void VisualSettings::refresh ( ) [slot]
```

Upon clicking the refresh button load the values and show the graphics.

#### 14.37.3.6 saveAndExit

```
void VisualSettings::saveAndExit ( ) [slot]
```

Emit signal for saving the configuration and exit the dialog.

#### 14.37.3.7 saveSettings

```
void VisualSettings::saveSettings (
    QColor probandColor,
    QColor promotedColor,
    QColor standardColor,
    QColor highlightedColor,
    int sizeX,
    int sizeY,
    QFont font,
    QColor lineColor,
    QColor textColor,
    int lineWidth,
    int rounding ) [signal]
```

Signal to emit when OK button is pressed to save this configuration.

##### Parameters

|                         |                                    |
|-------------------------|------------------------------------|
| <i>probandColor</i>     | Color used for proband.            |
| <i>promotedColor</i>    | Color for promoted Relations.      |
| <i>standardColor</i>    | Standard color.                    |
| <i>highlightedColor</i> | Highlighted color.                 |
| <i>sizeX</i>            | Size of the box in x axis.         |
| <i>sizeY</i>            | Size of the box in y axis.         |
| <i>font</i>             | Used font for the text in the box. |
| <i>lineColor</i>        | Color for lines.                   |
| <i>textColor</i>        | Color for texts.                   |
| <i>lineWidth</i>        | Width of the lines.                |
| <i>rounding</i>         | Rounding of the boxes.             |

#### 14.37.3.8 standardColor

```
void VisualSettings::standardColor ( ) [slot]
```

Show dialog for color change upon clicking on standard color change.

### 14.37.3.9 textColor

```
void VisualSettings::textColor ( ) [slot]
```

Change the color of a text.

## 14.37.4 Member Data Documentation

### 14.37.4.1 borderRadius\_

```
int VisualSettings::borderRadius_ [private]
```

Border radius of the box.

### 14.37.4.2 font\_

```
QFont VisualSettings::font_ [private]
```

Used font.

### 14.37.4.3 highlightedColor\_

```
QColor VisualSettings::highlightedColor_ [private]
```

[Color](#) for highlighting the box.

### 14.37.4.4 pen\_

```
QPen VisualSettings::pen_ [private]
```

Which pen is used for lines.

### 14.37.4.5 probandColor\_

```
QColor VisualSettings::probandColor_ [private]
```

[Color](#) for proband.

### 14.37.4.6 promotedColor\_

```
QColor VisualSettings::promotedColor_ [private]
```

[Color](#) for promoted relations.

#### 14.37.4.7 scene

```
QGraphicsScene* VisualSettings::scene [private]
```

Used graphics scene.

#### 14.37.4.8 sizeX\_

```
int VisualSettings::sizeX_ [private]
```

Size of the box in x axis.

#### 14.37.4.9 sizeY\_

```
int VisualSettings::sizeY_ [private]
```

Size of the box in y axis.

#### 14.37.4.10 standardColor\_

```
QColor VisualSettings::standardColor_ [private]
```

[Color](#) for standard relations.

#### 14.37.4.11 textPen\_

```
QPen VisualSettings::textPen_ [private]
```

Pen used for texts.

#### 14.37.4.12 ui

```
Ui::VisualSettings* VisualSettings::ui [private]
```

Used Qt user interface.

The documentation for this class was generated from the following files:

- [graphics/dialogs.h](#)
- [graphics/dialogs.cpp](#)

## 14.38 WrappedDate Class Reference

Class holding two dates and date description. Always first date is the main one.

```
#include <date.h>
```

## Public Member Functions

- [WrappedDate](#) ()  
*Default constructor.*
- void [clear](#) ()  
*Clear all values.*
- const [Date](#) & [getFirstDate](#) () const
- const [Date](#) & [getSecondDate](#) () const
- const std::string & [getText](#) () const
- bool [isExact](#) () const
- bool [isUnknown](#) () const
- void [readJson](#) (const Json::Value &value)
- void [setFirstDate](#) (int y, int m, int d)
- void [setSecondDate](#) (int y, int m, int d)
- std::string & [setText](#) ()
- std::string [str](#) () const
- void [updateDate](#) ()  
*Switch the dates if the first one is empty.*

## Private Attributes

- [Date](#) [date1\\_](#)  
*Main date, if interval is set it is the first day.*
- [Date](#) [date2\\_](#)  
*Second date for interval only, otherwise empty.*
- std::string [text\\_](#)  
*If text was given.*

## 14.38.1 Detailed Description

Class holding two dates and date description. Always first date is the main one.

## 14.38.2 Constructor & Destructor Documentation

### 14.38.2.1 WrappedDate()

```
WrappedDate::WrappedDate ( )
```

Default constructor.

## 14.38.3 Member Function Documentation

### 14.38.3.1 clear()

```
void WrappedDate::clear ( )
```

Clear all values.



#### 14.38.3.2 getFirstDate()

```
const Date & WrappedDate::getFirstDate ( ) const
```

Get the first date.

##### Returns

Constant reference to the first date.

#### 14.38.3.3 getSecondDate()

```
const Date & WrappedDate::getSecondDate ( ) const
```

Get the second date.

##### Returns

Constant reference to the second date.

#### 14.38.3.4 getText()

```
const std::string & WrappedDate::getText ( ) const
```

Get the text.

##### Returns

Constant reference to the text.

#### 14.38.3.5 isExact()

```
bool WrappedDate::isExact ( ) const
```

Check if the date is exact or not. To be exact only the first date has to be exact, the second got to be empty.

##### Returns

True if the first date is exact & second empty.

#### 14.38.3.6 isUnknown()

```
bool WrappedDate::isUnknown ( ) const
```

Check if the date is unknown. To be unknown all values are 0 and text is empty.

##### Returns

If the date is unknown.

#### 14.38.3.7 readJson()

```
void WrappedDate::readJson (
    const Json::Value & value )
```

Read the data from JSON formatting.

**Parameters**

|              |                   |
|--------------|-------------------|
| <i>value</i> | Given JSON value. |
|--------------|-------------------|

**14.38.3.8 setFirstDate()**

```
void WrappedDate::setFirstDate (
    int y,
    int m,
    int d )
```

Set the first date.

**Parameters**

|          |               |
|----------|---------------|
| <i>y</i> | Year number.  |
| <i>m</i> | Month number. |
| <i>d</i> | Day number.   |

**14.38.3.9 setSecondDate()**

```
void WrappedDate::setSecondDate (
    int y,
    int m,
    int d )
```

Set the second date.

**Parameters**

|          |               |
|----------|---------------|
| <i>y</i> | Year number.  |
| <i>m</i> | Month number. |
| <i>d</i> | Day number.   |

**14.38.3.10 setText()**

```
std::string & WrappedDate::setText ( )
```

Set the text for the date by giving the reference to the text.

**Returns**

Text used in the date.

#### 14.38.3.11 str()

```
std::string WrappedDate::str ( ) const
```

Represent the date by a string.

##### Returns

Newly constructed string.

#### 14.38.3.12 updateDate()

```
void WrappedDate::updateDate ( )
```

Switch the dates if the first one is empty.

### 14.38.4 Member Data Documentation

#### 14.38.4.1 date1\_

```
Date WrappedDate::date1_ [private]
```

Main date, if interval is set it is the first day.

#### 14.38.4.2 date2\_

```
Date WrappedDate::date2_ [private]
```

Second date for interval only, otherwise empty.

#### 14.38.4.3 text\_

```
std::string WrappedDate::text_ [private]
```

If text was given.

The documentation for this class was generated from the following files:

- [core/date.h](#)
- [core/date.cpp](#)



# Chapter 15

## File Documentation

### 15.1 core/config.cpp File Reference

Source code for configuring the app such as special evens and relations.

```
#include "config.h"
```

#### Functions

- `std::ostream & operator<< (std::ostream &os, const EventTemplate &et)`
- `std::ostream & operator<< (std::ostream &os, const RelationTemplate &rt)`
- `std::ostream & operator<< (std::ostream &os, const Color &color)`
- `std::ostream & operator<< (std::ostream &os, const AppSettings &as)`
- `std::ostream & operator<< (std::ostream &os, const Settings &settings)`

#### 15.1.1 Detailed Description

Source code for configuring the app such as special evens and relations.

#### 15.1.2 Function Documentation

##### 15.1.2.1 `operator<<()` [1/5]

```
std::ostream & operator<< (  
    std::ostream & os,  
    const AppSettings & as )
```

Output the app settings to stream in JSON format.

#### Parameters

|           |                     |
|-----------|---------------------|
| <i>os</i> | Given stream.       |
| <i>as</i> | Given app settings. |

**Returns**

Reference to the changed stream.

**15.1.2.2 operator<<() [2/5]**

```
std::ostream & operator<< (
    std::ostream & os,
    const Color & color )
```

Output the color to stream in JSON format.

**Parameters**

|              |               |
|--------------|---------------|
| <i>os</i>    | Given stream. |
| <i>color</i> | Given color.  |

**Returns**

Reference to the changed stream.

**15.1.2.3 operator<<() [3/5]**

```
std::ostream & operator<< (
    std::ostream & os,
    const EventTemplate & et )
```

Output the event template to stream in JSON format.

**Parameters**

|           |                       |
|-----------|-----------------------|
| <i>os</i> | Given stream.         |
| <i>et</i> | Given event template. |

**Returns**

Reference to the changed stream.

**15.1.2.4 operator<<() [4/5]**

```
std::ostream & operator<< (
    std::ostream & os,
    const RelationTemplate & rt )
```

Output the relation template to stream in JSON format.

**Parameters**

|           |                          |
|-----------|--------------------------|
| <i>os</i> | Given stream.            |
| <i>rt</i> | Given relation template. |

**Returns**

Reference to the changed stream.

**15.1.2.5 operator<<() [5/5]**

```
std::ostream & operator<< (
    std::ostream & os,
    const Settings & setting )
```

Output all the settings to stream in JSON format.

**Parameters**

|                |                 |
|----------------|-----------------|
| <i>os</i>      | Given stream.   |
| <i>setting</i> | Given settings. |

**Returns**

Reference to the changed stream.

**15.2 core/config.h File Reference**

Header file for configuration of the app. Also includes templates for events and relations.

```
#include <vector>
#include <string>
#include <ostream>
#include <map>
#include <optional>
#include <json/json.h>
#include <climits>
#include <algorithm>
#include "strings.h"
```

**Classes**

- class [EventTemplate](#)  
*Template for events. It specifies whether the event includes a date, a place or if more persons are involved and if they have some special roles.*
- class [RelationTemplate](#)  
*Template for relation. What is the name, description. What is the generation difference. And how are persons in this relation called.*

- struct [Color](#)  
*Struct for general color.*
- struct [AppSettings](#)  
*Struct for app settings.*
- class [Settings](#)  
*Class holding all data about the settings of the app and all templates.*

## Namespaces

- namespace [trait](#)  
*Namespace for everything with relation traits.*

## Enumerations

- enum [Trait](#) {  
[None](#) , [Fatherhood](#) , [Motherhood](#) , [Partnership](#) ,  
[Sibling](#) }  
*Common traits of each relations.*
- enum [TabPosition](#) { [North](#) , [South](#) , [East](#) , [West](#) }  
*Enumeration for tab position.*

## Functions

- `std::ostream & operator<< (std::ostream &os, const EventTemplate &et)`
- `std::string trait::str (Trait t)`
- `std::ostream & operator<< (std::ostream &os, const RelationTemplate &rt)`
- `std::ostream & operator<< (std::ostream &os, const Color &color)`
- `std::ostream & operator<< (std::ostream &os, const AppSettings &as)`
- `std::ostream & operator<< (std::ostream &os, const Settings &setting)`

## Variables

- `constexpr Trait trait::AllTraits [] = {None, Fatherhood, Motherhood, Partnership, Sibling}`  
*All possible traits.*

## 15.2.1 Detailed Description

Header file for configuration of the app. Also includes templates for events and relations.

## 15.2.2 Enumeration Type Documentation

### 15.2.2.1 TabPosition

```
enum TabPosition
```

Enumeration for tab position.



**Enumerator**

|       |  |
|-------|--|
| North |  |
| South |  |
| East  |  |
| West  |  |

**15.2.2.2 Trait**

enum [Trait](#)

Common traits of each relations.

**Enumerator**

|             |  |
|-------------|--|
| None        |  |
| Fatherhood  |  |
| Motherhood  |  |
| Partnership |  |
| Sibling     |  |

**15.2.3 Function Documentation****15.2.3.1 operator<<() [1/5]**

```
std::ostream & operator<< (
    std::ostream & os,
    const AppSettings & as )
```

Output the app settings to stream in JSON format.

**Parameters**

|           |                     |
|-----------|---------------------|
| <i>os</i> | Given stream.       |
| <i>as</i> | Given app settings. |

**Returns**

Reference to the changed stream.

**15.2.3.2 operator<<() [2/5]**

```
std::ostream & operator<< (
    std::ostream & os,
    const Color & color )
```

Output the color to stream in JSON format.

**Parameters**

|              |               |
|--------------|---------------|
| <i>os</i>    | Given stream. |
| <i>color</i> | Given color.  |

**Returns**

Reference to the changed stream.

**15.2.3.3 operator<<() [3/5]**

```
std::ostream & operator<< (
    std::ostream & os,
    const EventTemplate & et )
```

Output the event template to stream in JSON format.

**Parameters**

|           |                       |
|-----------|-----------------------|
| <i>os</i> | Given stream.         |
| <i>et</i> | Given event template. |

**Returns**

Reference to the changed stream.

**15.2.3.4 operator<<() [4/5]**

```
std::ostream & operator<< (
    std::ostream & os,
    const RelationTemplate & rt )
```

Output the relation template to stream in JSON format.

**Parameters**

|           |                          |
|-----------|--------------------------|
| <i>os</i> | Given stream.            |
| <i>rt</i> | Given relation template. |

**Returns**

Reference to the changed stream.

**15.2.3.5 operator<<() [5/5]**

```
std::ostream & operator<< (
    std::ostream & os,
    const Settings & setting )
```

Output all the settings to stream in JSON format.

#### Parameters

|                |                 |
|----------------|-----------------|
| <i>os</i>      | Given stream.   |
| <i>setting</i> | Given settings. |

#### Returns

Reference to the changed stream.

## 15.3 config.h

[Go to the documentation of this file.](#)

```

00001
00002 #ifndef config_h_
00003 #define config_h_
00004
00005 #include <vector>
00006 #include <string>
00007 #include <ostream>
00008 #include <map>
00009 #include <optional>
00010 #include <json/json.h>
00011 #include <climits>
00012 #include <algorithm>
00013 #include "strings.h"
00014
00016 class EventTemplate{
00017     public:
00019         EventTemplate();
00022         bool containsDate() const;
00025         bool containsPlace() const;
00028         const std::string& getDescription() const;
00031         size_t getId() const;
00034         const std::vector<std::string>& getRoles() const;
00037         const std::string& getTitle() const;
00040         bool hasMorePeopleInvolved() const;
00043         bool hasRoles() const;
00046         void readJson(const Json::Value& value);
00049         void setContainsDate(bool hasDate);
00052         void setContainsPlace(bool hasPlace);
00055         std::string& setDescription();
00058         void setId(size_t id);
00061         void setMorePeopleInvolved(bool morePeople);
00064         void setRoles(std::vector<std::string>& roles);
00067         std::string& setTitle();
00070         std::string str() const;
00071     private:
00073         bool containsDate_;
00075         bool containsPlace_;
00077         std::string description_;
00079         size_t id_;
00081         bool morePeopleInvolved_;
00083         std::vector<std::string> roles_;
00085         std::string title_;
00086 };
00087
00092 std::ostream& operator<<(std::ostream& os, const EventTemplate& et);
00093
00095 enum Trait {None, Fatherhood, Motherhood, Partnership, Sibling};
00096
00098 namespace trait{
00100     constexpr Trait AllTraits[] = {None, Fatherhood, Motherhood, Partnership, Sibling};
00104     std::string str(Trait t);
00105 }
00106
00108 class RelationTemplate{
00109     public:
00111         RelationTemplate();
00114         const std::string& getDescription() const;
00117         const std::string& getFirstName() const;
00120         int getGenerationDifference() const;
00123         size_t getId() const;
00126         const std::string& getSecondName() const;

```

```

00129         const std::string& getTitle() const;
00132         Trait getTrait() const;
00135         void readJson(const Json::Value& value);
00138         std::string& setDescription();
00141         std::string& setFirstName();
00144         void setGenerationDifference(int diff);
00147         void setId(size_t id);
00150         std::string& setSecondName();
00153         std::string& setTitle();
00156         void setTrait(Trait trait);
00159         std::string str() const;
00160     private:
00162         std::string description_;
00164         std::string firstName_;
00166         int generationDiff_;
00168         size_t id_;
00170         std::string secondName_;
00172         std::string title_;
00174         Trait trait_;
00175 };
00176
00181 std::ostream& operator<<(std::ostream& os, const RelationTemplate& rt);
00182
00184 enum TabPosition {North, South, East, West};
00185
00187 struct Color{
00192     explicit Color(int r = 255, int g = 255, int b = 255);
00194     int red;
00196     int green;
00198     int blue;
00200     void readJson(const Json::Value& value);
00201 };
00202
00207 std::ostream& operator<<(std::ostream& os, const Color& color);
00208
00210 struct AppSettings{
00212     std::string fontFamily;
00214     int fontSize;
00216     int genSizeDown;
00218     int genSizeUp;
00220     Color highlightedColor;
00222     Color lineColor;
00224     int lineWidth;
00226     Color probandColor;
00228     Color promoted;
00230     int radius;
00233     void readJson(const Json::Value& value);
00235     void setDefault();
00237     bool showEventTab;
00239     bool showFileTab;
00241     bool showGeneralTab;
00243     bool showMediaTab;
00245     bool showNoteTab;
00247     bool showRelationTab;
00249     int sizeX;
00251     int sizeY;
00253     int splitterPositionOne;
00255     int splitterPositionTwo;
00257     Color standard;
00259     TabPosition tabPosition;
00261     Color textColor;
00262 };
00263
00268 std::ostream& operator<<(std::ostream& os, const AppSettings& as);
00269
00271 class Settings{
00272     public:
00274         Settings();
00276         void addDefaultEvents();
00278         void addDefaultRelations();
00282         size_t addEventTemplate(EventTemplate& et);
00286         size_t addRelationTemplate(RelationTemplate& rt);
00289         void addTag(const std::string& tag);
00291         void clear();
00296         void exportTemplates(std::ostream& os, const std::vector<size_t>& relIds, const
std::vector<size_t>& eventIds);
00299         const AppSettings& getAppSettings() const;
00303         std::optional<EventTemplate*> getEventTemplate(size_t id) const;
00306         const std::map<size_t, std::unique_ptr<EventTemplate>&& getEventTemplates() const;
00309         size_t getGlobalMainPerson() const;
00313         std::optional<RelationTemplate*> getRelationTemplate(size_t id) const;
00316         const std::map<size_t, std::unique_ptr<RelationTemplate>&& getRelationTemplates() const;
00319         const std::vector<std::string>& getTags() const;
00323         size_t importTemplates(const Json::Value& value);
00326         std::pair<size_t, size_t> indexes();
00329         void readJson(const Json::Value& value);
00333         void reIndex(size_t plusEvent, size_t plusRel);

```

```

00336         void removeEventTemplate(size_t index);
00339         void removeRelationTemplate(size_t index);
00342         void removeTag(const std::string& tag);
00345         AppSettings& setAppSettings();
00348         void setGlobalMainPerson(size_t id);
00349     private:
00357         void addDefaultEvent(const std::string& title, const std::string& description, bool date, bool
place, bool morePeople, std::vector<std::string> roles);
00365         void addDefaultRelation(const std::string& title, const std::string& description, const
std::string& firstName, const std::string& secondName, int genDiff, Trait trait);
00367         AppSettings appSettings_;
00369         size_t eventTemplateIndex_;
00371         std::map<size_t, std::unique_ptr<EventTemplate>> eventTemplates_;
00373         size_t globalMainPerson_;
00375         size_t relationTemplateIndex_;
00377         std::map<size_t, std::unique_ptr<RelationTemplate>> relationsTemplates_;
00379         std::vector<std::string> tags_;
00380     };
00381
00386 std::ostream& operator<<(std::ostream& os, const Settings& setting);
00387
00388 #endif

```

## 15.4 core/date.cpp File Reference

Source file for date formats.

```
#include "date.h"
```

### Functions

- `std::ostream & operator<<(std::ostream &os, const Date &d)`
- `std::ostream & operator<<(std::ostream &os, const WrappedDate &wd)`

### 15.4.1 Detailed Description

Source file for date formats.

### 15.4.2 Function Documentation

#### 15.4.2.1 operator<<() [1/2]

```

std::ostream & operator<<(
    std::ostream & os,
    const Date & d )

```

Writing date to the stream in format yyyy-mm-dd. Instead of writing for example 05, there will be just 5.

#### Parameters

|           |               |
|-----------|---------------|
| <i>os</i> | Given stream. |
| <i>d</i>  | Given date.   |

**Returns**

Reference to the changed stream.

**15.4.2.2 operator<<() [2/2]**

```
std::ostream & operator<< (
    std::ostream & os,
    const WrappedDate & wd )
```

Writing wrapped date to given stream in JSON formatting.

**Parameters**

|           |                                     |
|-----------|-------------------------------------|
| <i>os</i> | Given stream.                       |
| <i>wd</i> | Given <a href="#">WrappedDate</a> . |

**Returns**

Reference to the changed stream.

**15.5 core/date.h File Reference**

Header file for date formats.

```
#include <string>
#include <ostream>
#include <sstream>
#include <json/json.h>
#include <ctime>
#include "strings.h"
```

**Classes**

- class [Date](#)  
*Basic date format. Holding three numbers for year, month and day. If some of the numbers are not defined 0 is present.*
- class [WrappedDate](#)  
*Class holding two dates and date description. Always first date is the main one.*

**Functions**

- std::ostream & [operator<<](#) (std::ostream &os, const [Date](#) &d)
- std::ostream & [operator<<](#) (std::ostream &os, const [WrappedDate](#) &wd)

**15.5.1 Detailed Description**

Header file for date formats.

## 15.5.2 Function Documentation

### 15.5.2.1 operator<<() [1/2]

```
std::ostream & operator<< (
    std::ostream & os,
    const Date & d )
```

Writing date to the stream in format yyyy-mm-dd. Instead of writing for example 05, there will be just 5.

#### Parameters

|           |               |
|-----------|---------------|
| <i>os</i> | Given stream. |
| <i>d</i>  | Given date.   |

#### Returns

Reference to the changed stream.

### 15.5.2.2 operator<<() [2/2]

```
std::ostream & operator<< (
    std::ostream & os,
    const WrappedDate & wd )
```

Writing wrapped date to given stream in JSON formatting.

#### Parameters

|           |                                  |
|-----------|----------------------------------|
| <i>os</i> | Given stream.                    |
| <i>wd</i> | Given <code>WrappedDate</code> . |

#### Returns

Reference to the changed stream.

## 15.6 date.h

[Go to the documentation of this file.](#)

```
00001
00002 #ifndef date_h_
00003 #define date_h_
00004
00005 #include <string>
00006 #include <ostream>
00007 #include <sstream>
00008 #include <json/json.h>
00009 #include <ctime>
00010 #include "strings.h"
00011
00013 class Date{
00014     public:
00019         explicit Date(int y = 0, int m = 0, int d = 0);
00022         int getDay() const;
```

```

00025         int getMonth() const;
00028         int getYear() const;
00032         bool isEmpty() const;
00036         bool isExact() const;
00041         void read(const std::string& date);
00044         void setDay(int d);
00047         void setMonth(int m);
00050         void setYear(int y);
00054         std::string str() const;
00055     private:
00057         int day;
00059         int month;
00061         int year;
00062 };
00063
00069 std::ostream& operator<<(std::ostream& os, const Date& d);
00070
00072 class WrappedDate{
00073     public:
00075         WrappedDate();
00077         void clear();
00080         const Date& getFirstDate() const;
00083         const Date& getSecondDate() const;
00086         const std::string& getText() const;
00090         bool isExact() const;
00093         bool isUnknown() const;
00096         void readJson(const Json::Value& value);
00101         void setFirstDate(int y, int m, int d);
00106         void setSecondDate(int y, int m, int d);
00109         std::string& setText();
00112         std::string str() const;
00114         void updateDate();
00115     private:
00117         Date date1_;
00119         Date date2_;
00121         std::string text_;
00122 };
00123
00128 std::ostream& operator<<(std::ostream& os, const WrappedDate& wd);
00129
00130 #endif

```

## 15.7 core/family\_tree.cpp File Reference

Source file for Family tree.

```
#include "family_tree.h"
```

### 15.7.1 Detailed Description

Source file for Family tree.

## 15.8 core/family\_tree.h File Reference

Header file for Family tree.

```

#include <string>
#include <sstream>
#include <vector>
#include <ostream>
#include <map>
#include <set>
#include <memory>

```



```
#include <functional>
#include <json/json.h>
#include "file_parser.h"
#include "config.h"
#include "date.h"
#include "person.h"
#include "strings.h"
#include "family_tree_items.h"
```

## Classes

- class [FamilyTree](#)

*Main class for holding all data of a family tree. Also it acts as a bridge to the core of the application.*

### 15.8.1 Detailed Description

Header file for Family tree.

## 15.9 family\_tree.h

[Go to the documentation of this file.](#)

```
00001
00002 #ifndef family_tree_
00003 #define family_tree_
00004
00005 #include <string>
00006 #include <sstream>
00007 #include <vector>
00008 #include <ostream>
00009 #include <map>
00010 #include <set>
00011 #include <memory>
00012 #include <functional>
00013 #include <json/json.h>
00014 #include "file_parser.h"
00015 #include "config.h"
00016 #include "date.h"
00017 #include "person.h"
00018 #include "strings.h"
00019 #include "family_tree_items.h"
00020
00021 class FamilyTree{
00022 public:
00023     FamilyTree();
00024     Event* addEvent();
00025     Event* addEvent(size_t index);
00026     File* addFile(const std::string& name, FileType type = GENERAL_FILE);
00027     File* addFile(const std::string& name, size_t index, FileType type = GENERAL_FILE);
00028     Person* addPerson();
00029     Person* addPerson(size_t index);
00030     Relation* addRelation();
00031     Relation* addRelation(size_t index);
00032     void append(FamilyTree& other);
00033     std::pair<bool, std::string> checkFileConsistence();
00034     void clear();
00035     void clearProjectPaths();
00036     size_t copyFile(const std::string& filePath, FileType type = GENERAL_FILE);
00037     void createDatabase(const std::string& dirPath);
00038     size_t createEmptyFile(const std::string& name, FileType type = GENERAL_FILE);
00039     void enableHelpOnStartup(bool help);
00040     void exportProject(const std::string& filePath, bool includingEvents, bool
00041 includingRelations);
00042     std::string createResourcesDir(const std::string& target);
00043     void exportTemplates(const std::vector<size_t>& eventTemplates, const std::vector<size_t>&
00044 relTemplates, const std::string& filename);
00045     std::optional<Event*> getEvent(size_t id);
00046     const std::map<size_t, std::unique_ptr<Event>& getEvents() const;
```

```

00110     std::optional<File*> getFile(size_t id, FileType type = GENERAL_FILE);
00115     void getFileAbsolutePath(std::string& absolutePath, size_t id, FileType type = GENERAL_FILE);
00119     const std::map<size_t, std::unique_ptr<File*>& getFiles(FileType type = GENERAL_FILE);
00122     std::vector<std::string> getOrphanFiles();
00125     Person* getMainPerson();
00128     const std::map<size_t, std::unique_ptr<Person*>& getPersons();
00132     std::optional<Person*> getPerson(size_t id);
00135     const std::vector<std::string>& getProjectPaths() const;
00139     std::optional<Relation*> getRelation(size_t id);
00142     const std::map<size_t, std::unique_ptr<Relation*>& getRelations();
00146     std::vector<RelationSuggestion> getRelationSuggestions(size_t relId);
00149     Settings* getSettings();
00154     void getUrlFile(size_t id, std::string& fileUrl, FileType type = GENERAL_FILE);
00158     bool importProject(const std::string& filePath);
00162     bool importRootDir(const std::string& dirPath);
00166     std::pair<bool, size_t> importTemplates(const std::string& filePath);
00169     bool isDirectorySet();
00172     bool isSaved() const;
00175     void log(const std::string& error);
00181     std::pair<bool, bool> openDatabase(const std::string& dirPath, std::string& errorMessage,
std::string& backupFile);
00185     bool openDatabase(std::string& errorMessage);
00189     bool openSingleFileDatabase(const std::string& filePath);
00192     void printHtml(const std::string& dirPath);
00195     void printMainPersonHtml(const std::string& outputFile);
00198     void promoteRelation(size_t id);
00200     void removeBackup();
00203     void removeEvent(size_t id);
00206     void removeEventTemplate(size_t id);
00209     size_t removeOrphanFiles();
00211     void removePerson();
00214     void removeRelation(size_t id);
00217     void removeRelationTemplate(size_t id);
00222     void renameFile(size_t id, const std::string& newFilename, FileType type);
00227     bool restoreBackup(const std::string& backupFile, std::string& errorMessage);
00230     void setMainPerson(Person* p);
00233     void setMainPerson(size_t id);
00239     void setRelation(size_t relId, size_t pers1Id, size_t pers2Id, size_t templId);
00241     void setUnsaved();
00244     bool showHelpOnStartup();
00247     bool storeDatabase();
00251     size_t templatesBasedOnEventTemplate(size_t templateId);
00255     size_t templatesBasedOnRelationTemplate(size_t templateId);
00258     void updateEventsWithTemplate(size_t templ);
00262     bool writeConfig(const std::string& newPath);
00263     private:
00265         std::map<size_t, std::unique_ptr<Event*>& allEvents_;
00267         std::map<size_t, std::unique_ptr<File*>& allFiles_;
00269         std::map<size_t, std::unique_ptr<File*>& allMedia_;
00271         std::map<size_t, std::unique_ptr<File*>& allNotes_;
00273         std::map<size_t, std::unique_ptr<Person*>& allPersons_;
00275         std::map<size_t, std::unique_ptr<Relation*>& allRelations_;
00279         bool checkFileTypeConsistence(std::ostream& os, FileType type);
00281         size_t event_index_;
00283         size_t file_index_;
00288         void getGeneralOrphaFiles(std::vector<std::string>& files, const std::map<size_t,
std::unique_ptr<File*>& container, FileType type);
00297         void getParentSuggestions(std::vector<RelationSuggestion>& suggestions, const Person* parent,
const Person* child, size_t id, Trait trait,
00298             const std::set<size_t>& forbiddenPersonParent, const std::set<size_t>&
forbiddenPersonChild);
00305         void getPartnersSuggestions(std::vector<RelationSuggestion>& suggestions, const Person*
person, const Person* second, size_t id, const std::set<size_t>& forbiddenPersons);
00312         void getSiblingsSuggestions(std::vector<RelationSuggestion>& suggestions, const Person*
person, const Person* second, size_t id, const std::set<size_t>& forbiddenPersons);
00318         bool loadDatabaseFile(const std::string& file, std::function<void(const Json::Value&)> reader,
std::string& errorMessage);
00320         Person* mainPerson_;
00322         size_t media_index_;
00324         size_t note_index_;
00326         bool openHelp;
00328         Parser parser_;
00330         size_t person_index_;
00333         void printCss(const std::string& dirPath);
00338         void printFiles(std::ostream& os, const std::map<size_t, std::unique_ptr<File*>& container,
const std::string& label);
00341         void printEvents(std::ostream& os);
00346         void printHtmlPerson(std::ostream& os, bool links, Person* p);
00349         void printPeople(std::ostream& os);
00352         void printRelations(std::ostream& os);
00354         std::vector<std::string> projectPaths;
00357         void readJsonEvents(const Json::Value& value);
00363         void readJsonFiles(const Json::Value& value, std::map<size_t, std::unique_ptr<File*>&
container, const std::string& label, size_t& index);
00366         void readJsonPeople(const Json::Value& value);
00369         void readJsonRelations(const Json::Value& value);
00371         size_t relation_index_;

```

```

00377         size_t removeGeneralOrphanFile(std::map<size_t, std::unique_ptr<File>& container, FileType
        type, const std::string& dir);
00379         bool savedChanges_;
00381         Settings settings_;
00384         void storeFileDatabase(FileType type);
00385     };
00386
00387 #endif

```

## 15.10 core/family\_tree\_items.cpp File Reference

Source file for classes closely related to family tree.

```
#include "family_tree_items.h"
```

### Functions

- std::ostream & [operator<<](#) (std::ostream &os, const [Event](#) &e)
- std::ostream & [operator<<](#) (std::ostream &os, const [Relation](#) &r)
- std::ostream & [operator<<](#) (std::ostream &os, const [File](#) &f)

### 15.10.1 Detailed Description

Source file for classes closely related to family tree.

### 15.10.2 Function Documentation

#### 15.10.2.1 [operator<<\(\)](#) [1/3]

```

std::ostream & operator<< (
    std::ostream & os,
    const Event & e )

```

Using << operator on event to print event in JSON formatting.

#### Parameters

|           |                      |
|-----------|----------------------|
| <i>os</i> | Given output stream. |
| <i>e</i>  | Given event.         |

#### Returns

Changed output stream.

#### 15.10.2.2 [operator<<\(\)](#) [2/3]

```

std::ostream & operator<< (
    std::ostream & os,
    const File & f )

```

Output file to JSON format.

#### Parameters

|           |               |
|-----------|---------------|
| <i>os</i> | Given stream. |
| <i>f</i>  | Given file.   |

#### Returns

Changed stream.

### 15.10.2.3 `operator<<()` [3/3]

```
std::ostream & operator<< (
    std::ostream & os,
    const Relation & r )
```

Output the relation in JSON format.

#### Parameters

|           |                      |
|-----------|----------------------|
| <i>os</i> | Given output stream. |
| <i>r</i>  | Given relation.      |

#### Returns

Changed output stream.

## 15.11 `core/family_tree_items.h` File Reference

Header file for classes closely related to family tree such as [Event](#), [Relation](#) and [File](#).

```
#include <string>
#include <sstream>
#include <vector>
#include <ostream>
#include <json/json.h>
#include "config.h"
#include "date.h"
#include "strings.h"
```

#### Classes

- class [Event](#)  
*Class holding data for a single event.*
- class [Relation](#)  
*Class holding data for a relationship.*
- struct [RelationSuggestion](#)  
*Struct for relation suggestions.*
- class [File](#)  
*Meta-data about one file (media or note). Its real path is [id]-[filename].*

## Functions

- `std::ostream & operator<< (std::ostream &os, const Event &e)`
- `std::ostream & operator<< (std::ostream &os, const Relation &r)`
- `std::ostream & operator<< (std::ostream &os, const File &f)`

### 15.11.1 Detailed Description

Header file for classes closely related to family tree such as [Event](#), [Relation](#) and [File](#).

### 15.11.2 Function Documentation

#### 15.11.2.1 `operator<<()` [1/3]

```
std::ostream & operator<< (  
    std::ostream & os,  
    const Event & e )
```

Using `<<` operator on event to print event in JSON formatting.

##### Parameters

|           |                      |
|-----------|----------------------|
| <i>os</i> | Given output stream. |
| <i>e</i>  | Given event.         |

##### Returns

Changed output stream.

#### 15.11.2.2 `operator<<()` [2/3]

```
std::ostream & operator<< (  
    std::ostream & os,  
    const File & f )
```

Output file to JSON format.

##### Parameters

|           |               |
|-----------|---------------|
| <i>os</i> | Given stream. |
| <i>f</i>  | Given file.   |

##### Returns

Changed stream.

### 15.11.2.3 operator<<() [3/3]

```
std::ostream & operator<< (
    std::ostream & os,
    const Relation & r )
```

Output the relation in JSON format.

#### Parameters

|           |                      |
|-----------|----------------------|
| <i>os</i> | Given output stream. |
| <i>r</i>  | Given relation.      |

#### Returns

Changed output stream.

## 15.12 family\_tree\_items.h

[Go to the documentation of this file.](#)

```
00001
00002 #ifndef family_tree_items_
00003 #define family_tree_items_
00004
00005 #include <string>
00006 #include <sstream>
00007 #include <vector>
00008 #include <ostream>
00009 #include <json/json.h>
00010 #include "config.h"
00011 #include "date.h"
00012 #include "strings.h"
00013
00015 class Event{
00016     public:
00019         explicit Event(Settings* settings);
00023         void addPerson(size_t person, const std::string& role = events::NO_ROLE);
00025         void clearPersons();
00028         WrappedDate* date();
00031         const WrappedDate& getDate() const;
00034         size_t getId() const;
00037         const std::vector<std::pair<std::string, size_t>& getPersons() const;
00040         const std::string& getPlace() const;
00043         const Settings& getSettings() const;
00046         size_t getTemplate() const;
00049         const std::string& getText() const;
00053         void printHtml(std::ostream& os, const std::vector<std::pair<std::string, std::string>&
persons);
00056         void readJson(const Json::Value& value);
00060         bool removePerson(size_t id);
00063         WrappedDate& setDate();
00066         void setId(size_t id);
00069         void setPlace(const std::string& place);
00072         void setTemplate(size_t templ);
00075         void setText(const std::string& text);
00078         std::string str() const;
00081         std::vector<size_t> updateToTemplate();
00082     private:
00084         WrappedDate date_;
00086         size_t id_;
00088         std::vector<std::pair<std::string, size_t> persons_;
00090         std::string place_;
00092         Settings* settings_;
00094         size_t template_;
00096         std::string text_;
00097
00098 };
00099
00104 std::ostream& operator <<(std::ostream& os, const Event& e);
00105
00107 class Relation{
```

```

00108     public:
00111         explicit Relation(Settings* settings);
00114         size_t getFirstPerson() const;
00117         size_t getId() const;
00120         size_t getSecondPerson() const;
00123         size_t getTemplate() const;
00126         const std::string& getTitle() const;
00130         size_t getTheOtherPerson(size_t current);
00134         const std::string& getTheOtherPersonName(size_t current);
00137         Trait getTrait();
00141         bool isOlderGeneration(size_t id);
00145         bool isParent(size_t personId);
00148         bool isSameGeneration();
00152         bool isYoungerGeneration(size_t id);
00158         void printHtml(std::ostream& os, const std::string& firstPerson, const std::string&
secondPerson, const std::string& promoted);
00161         void readJson(const Json::Value& value);
00164         void setId(size_t id);
00168         void setPersons(size_t person1, size_t person2);
00171         void setTemplate(size_t templ);
00176         std::string str(const std::string& firstPerson, const std::string& secondPerson) const;
00177     private:
00179         size_t id_;
00181         size_t person1_;
00183         size_t person2_;
00185         Settings* settings_;
00187         size_t template_;
00188 };
00189
00194 std::ostream& operator<<(std::ostream& os, const Relation& r);
00195
00197 struct RelationSuggestion{
00198     public:
00205         RelationSuggestion(size_t lockedP, size_t suggestedP, bool par, bool sym, Trait t);
00207         size_t lockedPerson;
00209         bool parent;
00211         size_t suggestedPerson;
00213         bool symmetrical;
00215         Trait trait;
00216 };
00217
00219 class File{
00220     public:
00222         File();
00225         const std::string& getFilename() const;
00228         size_t getId() const;
00231         std::string getRealName() const;
00234         void readJson(const Json::Value& value);
00237         void setFilename(const std::string& fileName);
00240         void setId(size_t id);
00241     private:
00243         std::string filename_;
00245         size_t id_;
00246 };
00247
00252 std::ostream& operator <<(std::ostream& os, const File& f);
00253
00254 #endif

```

## 15.13 core/file\_parser.cpp File Reference

Source file for working with predefined database.

```
#include "file_parser.h"
```

### 15.13.1 Detailed Description

Source file for working with predefined database.

## 15.14 core/file\_parser.h File Reference

The header file for working with the database and other files.

```
#include <filesystem>
#include <iostream>
#include <fstream>
#include <string>
#include <sstream>
#include <json/json.h>
#include <chrono>
#include <iomanip>
#include <ctime>
#include <set>
#include "strings.h"
```

### Classes

- class [Parser](#)  
*Main class for working with the database.*

### Namespaces

- namespace [parser](#)  
*Namespace for all strings representing files and directories in the database.*

### Variables

- const std::string [parser::FILES\\_DIR](#) = "Files"  
*Directory for all documents and files in general.*
- const std::string [parser::JSON\\_CONFIG](#) = "Config.json"  
*JSON file-name for configuration and templates.*
- const std::string [parser::JSON\\_EVENTS](#) = "Events.json"  
*JSON file-name for events.*
- const std::string [parser::JSON\\_FILES](#) = "Files.json"  
*JSON file-name for files meta-data.*
- const std::string [parser::JSON\\_MEDIA](#) = "Media.json"  
*JSON file-name for media meta-data.*
- const std::string [parser::JSON\\_NOTES](#) = "Notes.json"  
*JSON file-name for notes meta-data.*
- const std::string [parser::JSON\\_PERSONS](#) = "Persons.json"  
*JSON file-name for persons.*
- const std::string [parser::JSON\\_RELATIONS](#) = "Relations.json"  
*JSON file-name for relations.*
- const std::string [parser::MEDIA\\_DIR](#) = "Media"  
*Directory for all pictures and other media.*
- const std::string [parser::NOTES\\_DIR](#) = "Notes"  
*Directory for all notes.*
- const std::string [parser::ERROR\\_LOG](#) = ".error.log"  
*File for writing error log.*



## 15.14.1 Detailed Description

The header file for working with the database and other files.

## 15.15 file\_parser.h

[Go to the documentation of this file.](#)

```

00001
00002 #ifndef file_parser_h_
00003 #define file_parser_h_
00004
00005 #include <filesystem>
00006 #include <iostream>
00007 #include <fstream>
00008 #include <string>
00009 #include <sstream>
00010 #include <json/json.h>
00011 #include <chrono>
00012 #include <iomanip>
00013 #include <ctime>
00014 #include <set>
00015 #include "strings.h"
00016
00017 namespace parser{
00018     const std::string FILES_DIR = "Files";
00019     const std::string JSON_CONFIG = "Config.json";
00020     const std::string JSON_EVENTS = "Events.json";
00021     const std::string JSON_FILES = "Files.json";
00022     const std::string JSON_MEDIA = "Media.json";
00023     const std::string JSON_NOTES = "Notes.json";
00024     const std::string JSON_PERSONS = "Persons.json";
00025     const std::string JSON_RELATIONS = "Relations.json";
00026     const std::string MEDIA_DIR = "Media";
00027     const std::string NOTES_DIR = "Notes";
00028     const std::string ERROR_LOG = ".error.log";
00029 }
00030
00031 class Parser{
00032 public:
00033     Parser();
00034     void checkFileConsistency(std::set<std::string>& database, std::vector<std::string>& found,
00035 const std::string& dir);
00036     void clear();
00037     bool containsBackupFile(std::string& backupFile);
00038     bool copyFile(const std::string& filePath, const std::string& dir, std::string& fileName,
00039 size_t id);
00040     void createEmptyFile(const std::string& dir, const std::string& fileName);
00041     std::string createResourcesDir(const std::string& target);
00042     std::string getAbsolutePath(const std::string& dir, const std::string& fileName) const;
00043     void getFileInfo(const std::string& dir, const std::string& fileName, std::string& fileInfo)
00044 const;
00045     bool isRootDirectorySet();
00046     void log(const std::string& error);
00047     void makeNewDatabase(const std::string& dirPath);
00048     std::pair<bool, std::vector<std::string>> readConfig();
00049     bool readJSONFile(const std::string& fileName, Json::Value& root, bool inDatabase = true);
00050     void removeBackup();
00051     void removeFile(const std::string& filename, const std::string& dir);
00052     void renameFile(const std::string& originalName, const std::string& changedName, const
00053 std::string& dir);
00054     void restoreBackup(const std::string& backupFile);
00055     bool setDatabase(const std::string& dirPath);
00056     void writeConfig(bool help);
00057     void writeConfig(const std::vector<std::string>& rootPaths, bool help);
00058     void writeHtml(const std::string& html, const std::string& output);
00059     void writeHtml(const std::string& html, const std::string& dirPath, const std::string&
00060 filename);
00061     void writeJSON(const std::string& data, const std::string& file, bool inDatabase = true);
00062 private:
00063     bool containsBackup(std::string& backupFile, const std::string& filename);
00064     void createAllDirs();
00065     void createJsonFile(const std::string& fileName);
00066     std::filesystem::path root_;
00067     std::filesystem::path configPath_;
00068 };
00069
00070 #endif

```

## 15.16 core/person.cpp File Reference

Source file for all classes closely related to person.

```
#include "person.h"
```

### Functions

- `std::ostream & operator<< (std::ostream &os, const VirtualDrive &vd)`
- `void printSizeTContainer (std::ostream &os, const std::vector< size_t > &container, const std::string &label)`
- `std::ostream & operator<< (std::ostream &os, const Person &p)`

### 15.16.1 Detailed Description

Source file for all classes closely related to person.

### 15.16.2 Function Documentation

#### 15.16.2.1 operator<<() [1/2]

```
std::ostream & operator<< (
    std::ostream & os,
    const Person & p )
```

Write the data about the person in JSON format.

##### Parameters

|           |                      |
|-----------|----------------------|
| <i>os</i> | Is the given stream. |
| <i>p</i>  | Is the given person. |

##### Returns

Reference to the used output stream.

#### 15.16.2.2 operator<<() [2/2]

```
std::ostream & operator<< (
    std::ostream & os,
    const VirtualDrive & vd )
```

Print virtual drive to JSON format.

##### Parameters

|           |                      |
|-----------|----------------------|
| <i>os</i> | Given output stream. |
| <i>vd</i> | Given VirtualDrive.  |

#### Returns

Changed output stream.

#### 15.16.2.3 printSizeTContainer()

```
void printSizeTContainer (
    std::ostream & os,
    const std::vector< size_t > & container,
    const std::string & label )
```

Print to JSON formatting container of unsigned integers with given label.

#### Parameters

|                  |                              |
|------------------|------------------------------|
| <i>os</i>        | Given stream where to print. |
| <i>container</i> | Given container of ids.      |
| <i>label</i>     | The label of JSON array.     |

## 15.17 core/person.h File Reference

Header file for all classes closely related to person.

```
#include <string>
#include <vector>
#include <ostream>
#include <sstream>
#include <json/json.h>
#include "date.h"
#include "strings.h"
#include "config.h"
```

#### Classes

- class [VirtualDrive](#)  
*Class for virtualization of folder structure for each person.*
- class [Person](#)  
*One person in the family tree.*

#### Namespaces

- namespace [gender](#)  
*Namespace for constants with gender enumeration.*

## Enumerations

- enum `Gender` { `Other` , `Male` , `Female` }  
*Gender types: Male, Female and Other.*
- enum `FileType` { `GENERAL_FILE` , `MEDIA` , `NOTE` }  
*Enumeration on special types of files.*

## Functions

- std::ostream & `operator<<` (std::ostream &os, const `VirtualDrive` &vd)
- std::string `gender::str` (`Gender` g)
- void `printSizeTContainer` (std::ostream &os, const std::vector< size\_t > &container, const std::string &label)
- std::ostream & `operator<<` (std::ostream &os, const `Person` &p)

## Variables

- constexpr `Gender` `gender::AllGenders` [] = {`Other`, `Male`, `Female`}  
*All genders in a list.*

### 15.17.1 Detailed Description

Header file for all classes closely related to person.

### 15.17.2 Enumeration Type Documentation

#### 15.17.2.1 FileType

enum `FileType`

Enumeration on special types of files.

Enumerator

|              |  |
|--------------|--|
| GENERAL_FILE |  |
| MEDIA        |  |
| NOTE         |  |

#### 15.17.2.2 Gender

enum `Gender`

Gender types: Male, Female and Other.

Enumerator

|        |  |
|--------|--|
| Other  |  |
| Male   |  |
| Female |  |

## 15.17.3 Function Documentation

### 15.17.3.1 operator<<() [1/2]

```
std::ostream & operator<< (
    std::ostream & os,
    const Person & p )
```

Write the data about the person in JSON format.

#### Parameters

|           |                      |
|-----------|----------------------|
| <i>os</i> | Is the given stream. |
| <i>p</i>  | Is the given person. |

#### Returns

Reference to the used output stream.

### 15.17.3.2 operator<<() [2/2]

```
std::ostream & operator<< (
    std::ostream & os,
    const VirtualDrive & vd )
```

Print virtual drive to JSON format.

#### Parameters

|           |                      |
|-----------|----------------------|
| <i>os</i> | Given output stream. |
| <i>vd</i> | Given VirtualDrive.  |

#### Returns

Changed output stream.

### 15.17.3.3 printSizeTContainer()

```
void printSizeTContainer (
    std::ostream & os,
    const std::vector< size_t > & container,
    const std::string & label )
```

Print to JSON formatting container of unsigned integers with given label.

#### Parameters

|                  |                              |
|------------------|------------------------------|
| <i>os</i>        | Given stream where to print. |
| <i>container</i> | Given container of ids.      |
| <i>label</i>     | The label of JSON array.     |

## 15.18 person.h

[Go to the documentation of this file.](#)

```

00001
00002 #ifndef person_h_
00003 #define person_h_
00004
00005 #include <string>
00006 #include <vector>
00007 #include <ostream>
00008 #include <sstream>
00009 #include <json/json.h>
00010 #include "date.h"
00011 #include "strings.h"
00012 #include "config.h"
00013
00014 class VirtualDrive{
00015     public:
00016         VirtualDrive() = default;
00017         explicit VirtualDrive(const std::string& name);
00018         void addFile(size_t fileId);
00019         VirtualDrive* addSubdrive();
00020         VirtualDrive* addSubdrive(const std::string& name);
00021         void copyDrive(const VirtualDrive& other, size_t plusIndex = 0);
00022         const std::vector<size_t>& getFiles() const;
00023         const std::string& getName() const;
00024         const std::vector<std::unique_ptr<VirtualDrive>& getSubdrives() const;
00025         void readJson(const Json::Value& value);
00026         void remove(const std::string& subfolderName);
00027         void removeFile(size_t id);
00028         void rename(const std::string& name);
00029         void usedFiles(std::vector<size_t>& used) const;
00030     private:
00031         std::vector<size_t> files_;
00032         std::string name_;
00033         std::vector<std::unique_ptr<VirtualDrive> subdrives_;
00034 };
00035
00036 std::ostream& operator<<(std::ostream& os, const VirtualDrive& vd);
00037
00038 enum Gender {Other, Male, Female};
00039
00040 namespace gender{
00041     constexpr Gender AllGenders[] = {Other, Male, Female};
00042     std::string str(Gender g);
00043 }
00044
00045 enum FileType{GENERAL_FILE, MEDIA, NOTE};
00046
00047 class Person{
00048     public:
00049         Person();
00050         void addEvent(size_t id);
00051         void addRelation(size_t id);
00052         void addTag(const std::string& tag, const std::string& value);
00053         WrappedDate* birthDate();
00054         WrappedDate* deathDate();
00055         std::pair<bool, const std::string&> existsTag(const std::string& tag) const;
00056         void exportJson(std::ostream& os, bool includingEvents, bool includingRelations);
00057         const std::string& getAfterTitle() const;
00058         const WrappedDate& getBirthDate() const;
00059         const std::string& getBirthPlace() const;
00060         const WrappedDate& getDeathDate() const;
00061         const std::string& getDeathPlace() const;
00062         const std::vector<size_t>& getEvents() const;
00063         size_t getFather() const;
00064         std::string getFullName() const;
00065         const VirtualDrive& getFilesRoot(FileType type = GENERAL_FILE) const;
00066         VirtualDrive* getFilesRootPointer(FileType type);
00067         const std::string& getFrontTitle() const;
00068         Gender getGender() const;
00069         size_t getId() const;
00070         const std::string& getMaidenName() const;
00071         size_t getMother() const;
00072         const std::string& getName() const;
00073         size_t getPartner() const;
00074         const std::vector<size_t>& getRelations() const;
00075         const std::string& getSurname() const;
00076         const std::vector<std::pair<std::string, std::string>& getTags() const;
00077         void importJson(const Json::Value& value);
00078         bool isAlive() const;
00079         void printHtml(std::ostream& os);
00080         void promoteRelation(size_t id, Trait type);
00081         void readJson(const Json::Value& value);
00082         void removeEvent(size_t id);

```

```

00204     void removeRelation(size_t id);
00207     void removeTag(const std::string& tag);
00210     void setAfterTitle(const std::string& title);
00215     void setBirthDate(int y, int m, int d);
00218     std::string& setBirthPlace();
00223     void setDeathDate(int y, int m, int d);
00226     std::string& setDeathPlace();
00229     void setFather(size_t father);
00234     void setFilesRoot(const VirtualDrive& drive, FileType type = GENERAL_FILE, size_t plusIndex =
0);
00237     void setFrontTitle(const std::string& title);
00240     void setGender(Gender g);
00243     void setId(size_t id);
00246     void setLives(bool lives);
00249     std::string& setMaidenName();
00252     void setMother(size_t mother);
00255     std::string& setName();
00258     void setPartner(size_t id);
00261     std::string& setSurname();
00264     std::string str() const;
00268     void updateSpecialRelation(size_t relId, Trait trait);
00269 private:
00271     WrappedDate dateOfBirth_;
00273     WrappedDate dateOfDeath_;
00275     std::vector<size_t> events_;
00277     size_t father_;
00279     Gender gender_;
00281     size_t id_;
00283     bool lives_;
00285     std::string maidenName_;
00287     size_t mother_;
00289     std::string name_;
00291     size_t partner_;
00293     std::string placeOfBirth_;
00295     std::string placeOfDeath_;
00297     std::vector<size_t> relations_;
00299     VirtualDrive rootFileDrive_;
00301     VirtualDrive rootMediaDrive_;
00303     VirtualDrive rootNoteDrive_;
00305     std::string surname_;
00307     std::vector<std::pair<std::string, std::string>> tags_;
00309     std::string titleAfter_;
00311     std::string titleInFront_;
00312 };
00313
00318 void printSizeTContainer(std::ostream& os, const std::vector<size_t>& container, const std::string&
label);
00319
00324 std::ostream& operator<<(std::ostream& os, const Person& p);
00325
00326 #endif

```

## 15.19 core/strings.h File Reference

Is file containing all English strings.

```
#include <string>
```

### Namespaces

- namespace [person](#)  
*Namespace for strings related to a person.*
- namespace [tagchooser](#)  
*Namespace for strings shown in tag chooser dialog.*
- namespace [addfile](#)  
*Namespace for strings shown in add file dialog.*
- namespace [button](#)  
*Namespace for general texts on buttons.*
- namespace [error](#)

*Namespace for errors and exceptions messages.*

- namespace [relation](#)

*Namespace for strings for default relations.*

- namespace [events](#)

*Namespace for strings for default events.*

- namespace [html](#)

*Namespace for html tags and everything with html.*

- namespace [jsonlabel](#)

*Namespace for all JSON labels.*

## Variables

- const std::string [EMPTY\\_STRING](#) = ""

*Empty string.*

- const std::string [person::UNKNOWN\\_STR](#) = "Unknown"

*String for unknown tags.*

- const std::string [person::SURNAME\\_STR](#) = "Surname"

*String for default surname.*

- const std::string [person::NAME\\_STR](#) = "Name"

*String for default name.*

- const std::string [person::ROOT](#) = "root"

*String for default root directories.*

- const std::string [person::DATE\\_FROM](#) = "From: "

*String for date interval from.*

- const std::string [person::DATE\\_TO](#) = " to: "

*String for date interval to.*

- const std::string [tagchooser::CUSTOM\\_TAG](#) = "+ Custom tag"

*Last custom tag choice.*

- const std::string [addfile::NONE](#) = "None"

*String for no extension.*

- const std::string [button::ADD\\_NEW](#) = "Add new"

*Text for adding new element.*

- const std::string [button::CANCEL](#) = "Cancel"

*Text for cancelling.*

- const std::string [button::REMOVE](#) = "Remove"

*text for removing element.*

- const std::string [error::UNKNOWN\\_EXCEP\\_STR](#) = "Unknown exception catch."

*String for catching unknown exception.*

- const std::string [error::EXCEPTION\\_STR](#) = "Exception: "

*String for adding before exception description.*

- const std::string [error::CORRUPTED\\_DATABASE](#) = "Given database is corrupted: "

*String shown when database is corrupted.*

- const std::string [error::EMPTY\\_BOXES](#) = "All boxes must not be empty."

*String for showing that some boxes were left empty.*

- const std::string [error::NO\\_CHOSEN\\_FILE](#) = "No file was chosen."

*No file chosen and empty box.*

- const std::string [error::NO\\_ROLE](#) = "Given role 'no role' cannot be used. It would be considered as no special role."

*When person tries to add "no role" role, which is allocated for no roles.*

- const std::string [error::FORBIDDEN\\_CHARS](#) = "Text cannot contain forbidden characters. &lt; &gt; \\""



*When in any box where inserted forbidden characters.*

- `const std::string relation::FATHERHOOD = "Fatherhood"`  
*String for Fatherhood.*
- `const std::string relation::FATHERHOOD_DESC = "Relation between father and his child."`  
*String for Fatherhood description.*
- `const std::string relation::FATHER = "Father"`  
*String for father.*
- `const std::string relation::CHILD = "Child"`  
*String for child.*
- `const std::string relation::MOTHERHOOD = "Motherhood"`  
*String for Motherhood.*
- `const std::string relation::MOTHERHOOD_DESC = "Relation between mother and her child."`  
*String for Motherhood description.*
- `const std::string relation::MOTHER = "Mother"`  
*String for mother.*
- `const std::string relation::PARTNERSHIP = "Partnership"`  
*String for Partnership.*
- `const std::string relation::PARTNERSHIP_DESC = "Relation between two partners."`  
*String for description of partnership.*
- `const std::string relation::PARTNER = "Partner"`  
*String for partner.*
- `const std::string relation::MARRIAGE = "Marriage"`  
*String for Marriage.*
- `const std::string relation::MARRIAGE_DESC = "Bound by law between two people. Usually a man (husband) and a woman (wife)."`  
*String for Marriage description.*
- `const std::string relation::WIFE = "Wife"`  
*String for wife.*
- `const std::string relation::HUSBAND = "Husband"`  
*String for husband.*
- `const std::string relation::ADOPTIVE_FATHERHOOD = "Adoptive fatherhood"`  
*String for title for adoptive fatherhood.*
- `const std::string relation::DESC_ADOPTIVE_FATHERHOOD = "Fatherhood provided by adopting a child."`  
*String for description for adoptive fatherhood.*
- `const std::string relation::ADOPTIVE_FATHER = "Adoptive father"`  
*String for adoptive father.*
- `const std::string relation::ADOPTED_CHILD = "Adopted child"`  
*String for adopted child.*
- `const std::string relation::ADOPTIVE_MOTHERHOOD = "Adoptive motherhood"`  
*String for title for adoptive motherhood.*
- `const std::string relation::DESC_ADOPTIVE_MOTHERHOOD = "Motherhood provided by adopting a child."`  
*String for description for adoptive motherhood.*
- `const std::string relation::ADOPTIVE_MOTHER = "Adoptive mother"`  
*String for adoptive father.*
- `const std::string relation::BROTHERHOOD = "Brotherhood"`  
*String for brotherhood.*
- `const std::string relation::DESC_BROTHERHOOD = "Relation between two brothers."`  
*String for brotherhood description.*
- `const std::string relation::BROTHER = "Brother"`  
*String for brother.*
- `const std::string relation::SISTERHOOD = "Sisterhood"`

- String for sisterhood.*

  - `const std::string relation::DESC_SISTERHOOD = "Relation between two sisters."`

*String for sisterhood description.*
- `const std::string relation::SISTER = "Sister"`

*String for sister.*
- `const std::string relation::SIBLINGS = "Siblings"`

*String for siblings.*
- `const std::string relation::DESC_SIBLINGS = "Relation between brother and sister."`

*String for siblings description.*
- `const std::string events::WEDDING = "Wedding"`

*String for wedding.*
- `const std::string events::DESC_WEDDING = "A social event mainly between soon to be wife and husband."`

*String for description of wedding.*
- `const std::string events::GROOM = "Groom"`

*String for groom.*
- `const std::string events::BRIDE = "Bride"`

*String for bride.*
- `const std::string events::BEST_MAN = "Best man"`

*String for best man.*
- `const std::string events::MAID_OF_HONOR = "Maid of Honor"`

*String for maid of honor.*
- `const std::string events::DIVORCE = "Divorce"`

*String for divorce.*
- `const std::string events::DESC_DIVORCE = "Divorce is the process of terminating a marriage or marital union."`

*String for divorce description.*
- `const std::string events::FORMER_HUSBAND = "Former husband"`

*String for former husband.*
- `const std::string events::FORMER_WIFE = "Former wife"`

*String for former wife.*
- `const std::string events::FUNERAL = "Funeral"`

*String for funeral.*
- `const std::string events::DESC_FUNERAL = "A funeral is a ceremony connected with the final disposition of a corpse, such as a burial or cremation, with the attendant observances."`

*String for funeral description.*
- `const std::string events::DEAD_PERSON = "Dead person"`

*String for dead person.*
- `const std::string events::GRADUATION = "Graduation"`

*String for graduation.*
- `const std::string events::DESC_GRADUATION = "Graduation is the awarding of a diploma to a student by an educational institution."`

*String for graduation description.*
- `const std::string events::GRADUATE = "Graduate"`

*String for graduate.*
- `const std::string events::NO_ROLE = "no role"`

*String for no role in event.*
- `const std::string html::RESOURCES = "resources"`

*Where will resources be saved.*
- `const std::string html::PREFIX_CLOSE = "/close"`

*Prefix for close family diagram.*
- `const std::string html::PREFIX_FAMILY_TREE = "/tree"`

- Prefix for family tree diagram.*

  - const std::string [html::CUSTOM\\_CSS](#) = "styles.css"

*Custom CSS file name.*
- const std::string [html::GENERAL\\_INFO](#) = "General information"

*General information.*
- const std::string [html::BIRTH\\_DATE](#) = "Birth date: "

*Birth date.*
- const std::string [html::BIRTH\\_PLACE](#) = "Birth place: "

*Birth place.*
- const std::string [html::DEATH\\_DATE](#) = "Date of death: "

*Date of death.*
- const std::string [html::DEATH\\_PLACE](#) = "Place of death: "

*Place of death.*
- const std::string [html::GENDER](#) = "Gender: "

*Gender.*
- const std::string [html::CUSTOM\\_TAGS](#) = "Custom tags:"

*Custom tags.*
- const std::string [html::DESCRIPTION](#) = "Description:"

*Description.*
- const std::string [html::NOTE](#) = "Note:"

*Note.*
- const std::string [html::DATE](#) = "Date: "

*Date.*
- const std::string [html::PLACE](#) = "Place: "

*Place.*
- const std::string [html::PERSONS](#) = "Persons:"

*Persons.*
- const std::string [html::ALL\\_PERSONS](#) = "All persons"

*All persons.*
- const std::string [html::EVENTS](#) = "Events"

*Events.*
- const std::string [html::FAMILY\\_TREE](#) = "Family tree"

*Family tree.*
- const std::string [html::FAMILY\\_TREE\\_DIAGRAM](#) = "Family tree diagram"

*Family tree diagram.*
- const std::string [html::RELATIONS](#) = "Relations"

*Relations.*
- const std::string [html::CLOSE\\_FAMILY](#) = "Close family"

*Close family.*
- const std::string [html::CLOSE\\_FAMILY\\_DIAGRAM](#) = "Close family diagram"

*Close family diagram.*
- const std::string [jsonlabel::TITLE](#) = "title"

*JSON Label for title.*
- const std::string [jsonlabel::ID](#) = "id"

*JSON Label for id.*
- const std::string [jsonlabel::DESCRIPTION](#) = "description"

*JSON Label for description.*
- const std::string [jsonlabel::CONTAINS\\_DATE](#) = "contains date"

*JSON Label for if the event contains a date or not.*
- const std::string [jsonlabel::MORE\\_PEOPLE](#) = "more people"

*JSON Label for if the event contains more people.*

- `const std::string jsonlabel::SPECIAL_ROLES = "special roles"`  
*JSON Label for if the event has special roles.*
- `const std::string jsonlabel::NAME = "name"`  
*JSON Label for name.*
- `const std::string jsonlabel::FIRST_NAME = "first name"`  
*JSON Label for first name.*
- `const std::string jsonlabel::SECOND_NAME = "second name"`  
*JSON Label for second name.*
- `const std::string jsonlabel::TRAIT = "trait"`  
*JSON Label for trait.*
- `const std::string jsonlabel::GEN_DIFF = "generation difference"`  
*JSON Label for generation difference.*
- `const std::string jsonlabel::REL_TEMPL = "relation templates"`  
*JSON Label for relations templates.*
- `const std::string jsonlabel::EVE_TEMPL = "event templates"`  
*JSON Label for event templates.*
- `const std::string jsonlabel::TYPE = "type"`  
*JSON Label for type.*
- `const std::string jsonlabel::DATE = "date"`  
*JSON Label for date.*
- `const std::string jsonlabel::LAST_DATE = "last date"`  
*JSON Label for last date.*
- `const std::string jsonlabel::TEXT = "text"`  
*JSON Label for text.*
- `const std::string jsonlabel::TAGS = "tags"`  
*JSON Label for tags.*
- `const std::string jsonlabel::TAG = "tag"`  
*JSON Label for single tag.*
- `const std::string jsonlabel::RELATIONS = "relations"`  
*JSON Label for relations.*
- `const std::string jsonlabel::EVENTS = "events"`  
*JSON Label for events.*
- `const std::string jsonlabel::SURNAME = "surname"`  
*JSON Label for surname.*
- `const std::string jsonlabel::MAIDEN_NAME = "maiden name"`  
*JSON Label for maiden name.*
- `const std::string jsonlabel::VALUE = "value"`  
*JSON Label for value.*
- `const std::string jsonlabel::BIRTH = "birth"`  
*JSON Label for birth.*
- `const std::string jsonlabel::DEATH = "death"`  
*JSON Label for death.*
- `const std::string jsonlabel::VIRTUAL_DRIVE = "virtual drive"`  
*JSON Label for virtual drive.*
- `const std::string jsonlabel::MEDIA = "media"`  
*JSON Label for media.*
- `const std::string jsonlabel::NOTES = "notes"`  
*JSON Label for notes.*
- `const std::string jsonlabel::FILES = "files"`  
*JSON Label for files.*
- `const std::string jsonlabel::PERSONS = "persons"`

- JSON Label for persons.*

  - const std::string `jsonlabel::TEMPLATE` = "template"

*JSON Label for for template.*
- const std::string `jsonlabel::TEMPLATES` = "templates"

*JSON Label for for templates.*
- const std::string `jsonlabel::FIRST_PERSON` = "first person"

*JSON Label for first person.*
- const std::string `jsonlabel::SECOND_PERSON` = "second person"

*JSON Label for second person.*
- const std::string `jsonlabel::PLACE` = "place"

*JSON Label for place.*
- const std::string `jsonlabel::ROLE` = "role"

*JSON Label for role.*
- const std::string `jsonlabel::PERSON` = "person"

*JSON Label for person.*
- const std::string `jsonlabel::GENDER` = "gender"

*JSON Label for gender.*
- const std::string `jsonlabel::LIVES` = "lives"

*JSON Label for if the person lives or not.*
- const std::string `jsonlabel::FRONT_TITLE` = "title in front"

*JSON Label for title in front of a name.*
- const std::string `jsonlabel::AFTER_TITLE` = "title after"

*JSON Label for title after the name.*
- const std::string `jsonlabel::FATHER` = "father"

*JSON Label for father.*
- const std::string `jsonlabel::MOTHER` = "mother"

*JSON Label for mother.*
- const std::string `jsonlabel::PARTNER` = "partner"

*JSON Label for partner.*
- const std::string `jsonlabel::MOVEABLE_TABS` = "moveable tabs"

*JSON Label for setting moveable tabs.*
- const std::string `jsonlabel::TAB_POSITION` = "tab position"

*JSON Label for tab position.*
- const std::string `jsonlabel::GLOBAL_MAIN_PERSON` = "global main person"

*JSON Label for global main person.*
- const std::string `jsonlabel::CONTAINS_PLACE` = "contains place"

*JSON Label if an event template contains a place.*
- const std::string `jsonlabel::SHOW_GENERAL_TAB` = "show general tab"

*JSON Label for app settings if general tab should be visible.*
- const std::string `jsonlabel::SHOW_EVENT_TAB` = "show event tab"

*JSON Label for app settings if event tab should be visible.*
- const std::string `jsonlabel::SHOW_FILE_TAB` = "show file tab"

*JSON Label for app settings if file tab should be visible.*
- const std::string `jsonlabel::SHOW_MEDIA_TAB` = "show media tab"

*JSON Label for app settings if media tab should be visible.*
- const std::string `jsonlabel::SHOW_NOTE_TAB` = "show note tab"

*JSON Label for app settings if note tab should be visible.*
- const std::string `jsonlabel::SHOW_RELATION_TAB` = "show relation tab"

*JSON Label for app settings if relation tab should be visible.*
- const std::string `jsonlabel::APP_SETTINGS` = "app settings"

*JSON Label for app settings.*

- `const std::string jsonlabel::DRIVES = "drives"`  
*JSON Label for drives.*
- `const std::string jsonlabel::EXPORT = "export"`  
*JSON Label for export.*
- `const std::string jsonlabel::RED = "red"`  
*JSON Label for red.*
- `const std::string jsonlabel::BLUE = "blue"`  
*JSON Label for blue.*
- `const std::string jsonlabel::GREEN = "green"`  
*JSON Label for green.*
- `const std::string jsonlabel::FONT_FAMILY = "font family"`  
*JSON Label for font family.*
- `const std::string jsonlabel::FONT_SIZE = "font size"`  
*JSON Label for font size.*
- `const std::string jsonlabel::HIGHLIGHTED_COLOR = "highlighted color"`  
*JSON Label highlighted color.*
- `const std::string jsonlabel::PROBANT_COLOR = "proband color"`  
*JSON Label for proband color.*
- `const std::string jsonlabel::PROMOTED_COLOR = "promoted color"`  
*JSON Label promoted color.*
- `const std::string jsonlabel::STANDARD_COLOR = "standard color"`  
*JSON Label standard color.*
- `const std::string jsonlabel::X_SIZE = "x size"`  
*JSON Label for size in the x axis.*
- `const std::string jsonlabel::Y_SIZE = "y size"`  
*JSON Label for size in the y axis.*
- `const std::string jsonlabel::GEN_SIZE_UP = "number of generations up"`  
*JSON Label for how many generations are being shown upwards.*
- `const std::string jsonlabel::GEN_SIZE_DOWN = "number of generations down"`  
*JSON Label for how many generations are being shown downwards.*
- `const std::string jsonlabel::DIR = "dir"`  
*JSON Label for directory.*
- `const std::string jsonlabel::ROOT = "root"`  
*JSON Label for root.*
- `const std::string jsonlabel::HELP = "help"`  
*JSON Label for help.*
- `const std::string jsonlabel::SPLITTER_POSITION_ONE = "splitter position one"`  
*JSON Label for splitter position one.*
- `const std::string jsonlabel::SPLITTER_POSITION_TWO = "splitter position two"`  
*JSON Label for splitter position two.*
- `const std::string jsonlabel::TEXT_COLOR = "text color"`  
*JSON Label for the color of text.*
- `const std::string jsonlabel::LINE_COLOR = "line color"`  
*JSON Label for the line color.*
- `const std::string jsonlabel::LINE_WIDTH = "line width"`  
*JSON Label for line width.*
- `const std::string jsonlabel::BORDER_RADIUS = "radius"`  
*JSON Label border radius.*

## 15.19.1 Detailed Description

Is file containing all English strings.

## 15.19.2 Variable Documentation

### 15.19.2.1 EMPTY\_STRING

```
const std::string EMPTY_STRING = ""
```

Empty string.

## 15.20 strings.h

[Go to the documentation of this file.](#)

```
00001
00002 #ifndef strings_h_
00003 #define strings_h_
00004
00005 #include <string>
00006
00008 const std::string EMPTY_STRING = "";
00009
00010 // =====
00011 // Person strings
00012 // =====
00013
00015 namespace person{
00017     const std::string UNKNOWN_STR = "Unknown";
00019     const std::string SURNAME_STR = "Surname";
00021     const std::string NAME_STR = "Name";
00023     const std::string ROOT = "root";
00025     const std::string DATE_FROM = "From: ";
00027     const std::string DATE_TO = " to: ";
00028 }
00029
00030 // =====
00031 // Tag chooser strings
00032 // =====
00033
00035 namespace tagchooser{
00037     const std::string CUSTOM_TAG = "+ Custom tag";
00038 }
00039
00040 // =====
00041 // Add file
00042 // =====
00043
00045 namespace addfile{
00047     const std::string NONE = "None";
00048 }
00049
00050 // =====
00051 // General buttons texts.
00052 // =====
00053
00055 namespace button{
00057     const std::string ADD_NEW = "Add new";
00059     const std::string CANCEL = "Cancel";
00061     const std::string REMOVE = "Remove";
00062 }
00063
00064 // =====
00065 // Exceptions and errors strings.
00066 // =====
00067
00069 namespace error{
00071     const std::string UNKNOWN_EXCEP_STR = "Unknown exception catch.";
00073     const std::string EXCEPTION_STR = "Exception: ";
00075     const std::string CORRUPTED_DATABASE = "Given database is corrupted: ";
00077     const std::string EMPTY_BOXES = "All boxes must not be empty.";
00079     const std::string NO_CHOSEN_FILE = "No file was chosen.";
```

```

00081     const std::string NO_ROLE = "Given role 'no role' cannot be used. It would be considered as no
special role.";
00083     const std::string FORBIDDEN_CHARS = "Text cannot contain forbidden characters. &lt; &gt; \\";
00084 }
00085
00086 // =====
00087 // Default relations.
00088 // =====
00089
00091 namespace relation{
00093     const std::string FATHERHOOD = "Fatherhood";
00095     const std::string FATHERHOOD_DESC = "Relation between father and his child.";
00097     const std::string FATHER = "Father";
00099     const std::string CHILD = "Child";
00101     const std::string MOTHERHOOD = "Motherhood";
00103     const std::string MOTHERHOOD_DESC = "Relation between mother and her child.";
00105     const std::string MOTHER = "Mother";
00107     const std::string PARTNERSHIP = "Partnership";
00109     const std::string PARTNERSHIP_DESC = "Relation between two partners.";
00111     const std::string PARTNER = "Partner";
00113     const std::string MARRIAGE = "Marriage";
00115     const std::string MARRIAGE_DESC = "Bound by law between two people. Usually a man (husband) and a
woman (wife).";
00117     const std::string WIFE = "Wife";
00119     const std::string HUSBAND = "Husband";
00121     const std::string ADOPTIVE_FATHERHOOD = "Adoptive fatherhood";
00123     const std::string DESC_ADOPTIVE_FATHERHOOD = "Fatherhood provided by adopting a child.";
00125     const std::string ADOPTIVE_FATHER = "Adoptive father";
00127     const std::string ADOPTED_CHILD = "Adopted child";
00129     const std::string ADOPTIVE_MOTHERHOOD = "Adoptive motherhood";
00131     const std::string DESC_ADOPTIVE_MOTHERHOOD = "Motherhood provided by adopting a child.";
00133     const std::string ADOPTIVE_MOTHER = "Adoptive mother";
00135     const std::string BROTHERHOOD = "Brotherhood";
00137     const std::string DESC_BROTHERHOOD = "Relation between two brothers.";
00139     const std::string BROTHER = "Brother";
00141     const std::string SISTERHOOD = "Sisterhood";
00143     const std::string DESC_SISTERHOOD = "Relation between two sisters.";
00145     const std::string SISTER = "Sister";
00147     const std::string SIBLINGS = "Siblings";
00149     const std::string DESC_SIBLINGS = "Relation between brother and sister.";
00150 }
00151
00152 // =====
00153 // Default events.
00154 // =====
00155
00157 namespace events{
00159     const std::string WEDDING = "Wedding";
00161     const std::string DESC_WEDDING = "A social event mainly between soon to be wife and husband.;;
00163     const std::string GROOM = "Groom";
00165     const std::string BRIDE = "Bride";
00167     const std::string BEST_MAN = "Best man";
00169     const std::string MAID_OF_HONOR = "Maid of Honor";
00171     const std::string DIVORCE = "Divorce";
00173     const std::string DESC_DIVORCE = "Divorce is the process of terminating a marriage or marital
union.";
00175     const std::string FORMER_HUSBAND = "Former husband";
00177     const std::string FORMER_WIFE = "Former wife";
00179     const std::string FUNERAL = "Funeral";
00181     const std::string DESC_FUNERAL = "A funeral is a ceremony connected with the final disposition of
a corpse, such as a burial or cremation, with the attendant observances.";
00183     const std::string DEAD_PERSON = "Dead person";
00185     const std::string GRADUATION = "Graduation";
00187     const std::string DESC_GRADUATION = "Graduation is the awarding of a diploma to a student by an
educational institution.";
00189     const std::string GRADUATE = "Graduate";
00191     const std::string NO_ROLE = "no role";
00192 }
00193
00194 // =====
00195 // HTML Tags and everything with html.
00196 // =====
00197
00199 namespace html{
00201     const std::string RESOURCES = "resources";
00203     const std::string PREFIX_CLOSE = "/close-";
00205     const std::string PREFIX_FAMILY_TREE = "/tree-";
00207     const std::string CUSTOM_CSS = "styles.css";
00209     const std::string GENERAL_INFO = "General information";
00211     const std::string BIRTH_DATE = "Birth date: ";
00213     const std::string BIRTH_PLACE = "Birth place: ";
00215     const std::string DEATH_DATE = "Date of death: ";
00217     const std::string DEATH_PLACE = "Place of death: ";
00219     const std::string GENDER = "Gender: ";
00221     const std::string CUSTOM_TAGS = "Custom tags:";
00223     const std::string DESCRIPTION = "Description:";
00225     const std::string NOTE = "Note:";

```



```

00227     const std::string DATE = "Date: ";
00229     const std::string PLACE = "Place: ";
00231     const std::string PERSONS = "Persons:";
00233     const std::string ALL_PERSONS = "All persons";
00235     const std::string EVENTS = "Events";
00237     const std::string FAMILY_TREE = "Family tree";
00239     const std::string FAMILY_TREE_DIAGRAM = "Family tree diagram";
00241     const std::string RELATIONS = "Relations";
00243     const std::string CLOSE_FAMILY = "Close family";
00245     const std::string CLOSE_FAMILY_DIAGRAM = "Close family diagram";
00246 }
00247
00248 // =====
00249 // JSON Labels.
00250 // =====
00251
00253 namespace jsonlabel{
00255     const std::string TITLE = "title";
00257     const std::string ID = "id";
00259     const std::string DESCRIPTION = "description";
00261     const std::string CONTAINS_DATE = "contains date";
00263     const std::string MORE_PEOPLE = "more people";
00265     const std::string SPECIAL_ROLES = "special roles";
00267     const std::string NAME = "name";
00269     const std::string FIRST_NAME = "first name";
00271     const std::string SECOND_NAME = "second name";
00273     const std::string TRAIT = "trait";
00275     const std::string GEN_DIFF = "generation difference";
00277     const std::string REL_TEMPL = "relation templates";
00279     const std::string EVE_TEMPL = "event templates";
00281     const std::string TYPE = "type";
00283     const std::string DATE = "date";
00285     const std::string LAST_DATE = "last date";
00287     const std::string TEXT = "text";
00289     const std::string TAGS = "tags";
00291     const std::string TAG = "tag";
00293     const std::string RELATIONS = "relations";
00295     const std::string EVENTS = "events";
00297     const std::string SURNAME = "surname";
00299     const std::string MAIDEN_NAME = "maiden name";
00301     const std::string VALUE = "value";
00303     const std::string BIRTH = "birth";
00305     const std::string DEATH = "death";
00307     const std::string VIRTUAL_DRIVE = "virtual drive";
00309     const std::string MEDIA = "media";
00311     const std::string NOTES = "notes";
00313     const std::string FILES = "files";
00315     const std::string PERSONS = "persons";
00317     const std::string TEMPLATE = "template";
00319     const std::string TEMPLATES = "templates";
00321     const std::string FIRST_PERSON = "first person";
00323     const std::string SECOND_PERSON = "second person";
00325     const std::string PLACE = "place";
00327     const std::string ROLE = "role";
00329     const std::string PERSON = "person";
00331     const std::string GENDER = "gender";
00333     const std::string LIVES = "lives";
00335     const std::string FRONT_TITLE = "title in front";
00337     const std::string AFTER_TITLE = "title after";
00339     const std::string FATHER = "father";
00341     const std::string MOTHER = "mother";
00343     const std::string PARTNER = "partner";
00345     const std::string MOVEABLE_TABS = "moveable tabs";
00347     const std::string TAB_POSITION = "tab position";
00349     const std::string GLOBAL_MAIN_PERSON = "global main person";
00351     const std::string CONTAINS_PLACE = "contains place";
00353     const std::string SHOW_GENERAL_TAB = "show general tab";
00355     const std::string SHOW_EVENT_TAB = "show event tab";
00357     const std::string SHOW_FILE_TAB = "show file tab";
00359     const std::string SHOW_MEDIA_TAB = "show media tab";
00361     const std::string SHOW_NOTE_TAB = "show note tab";
00363     const std::string SHOW_RELATION_TAB = "show relation tab";
00365     const std::string APP_SETTINGS = "app settings";
00367     const std::string DRIVES = "drives";
00369     const std::string EXPORT = "export";
00371     const std::string RED = "red";
00373     const std::string BLUE = "blue";
00375     const std::string GREEN = "green";
00377     const std::string FONT_FAMILY = "font family";
00379     const std::string FONT_SIZE = "font size";
00381     const std::string HIGHLIGHTED_COLOR = "highlighted color";
00383     const std::string PROBANT_COLOR = "proband color";
00385     const std::string PROMOTED_COLOR = "promoted color";
00387     const std::string STANDARD_COLOR = "standard color";
00389     const std::string X_SIZE = "x size";
00391     const std::string Y_SIZE = "y size";
00393     const std::string GEN_SIZE_UP = "number of generations up";

```

```

00395     const std::string GEN_SIZE_DOWN = "number of generations down";
00397     const std::string DIR = "dir";
00399     const std::string ROOT = "root";
00401     const std::string HELP = "help";
00403     const std::string SPLITTER_POSITION_ONE = "splitter position one";
00405     const std::string SPLITTER_POSITION_TWO = "splitter position two";
00407     const std::string TEXT_COLOR = "text color";
00409     const std::string LINE_COLOR = "line color";
00411     const std::string LINE_WIDTH = "line width";
00413     const std::string BORDER_RADIUS = "radius";
00414 }
00415
00416 #endif

```

## 15.21 graphics/dialogs.cpp File Reference

Source file for implementing all dialogs.

```
#include "dialogs.h"
```

### 15.21.1 Detailed Description

Source file for implementing all dialogs.

## 15.22 graphics/dialogs.h File Reference

This file contains all window dialogs that will be shown. These classes are relatively small but there are many of them.

```

#include "ui_date_dialog.h"
#include "ui_tag_chooser.h"
#include "ui_add_file.h"
#include "ui_relation_templates.h"
#include "ui_event_templates.h"
#include "ui_event_dialog.h"
#include "ui_relation_dialog.h"
#include "ui_help_dialog.h"
#include "ui_choose_person.h"
#include "ui_visual_settings.h"
#include "ui_export_templates.h"
#include "ui_export_project.h"
#include "ui_html_export.h"
#include "ui_picture_export.h"
#include "ui_suggestions.h"
#include "graphics_items.h"
#include <QtCore>
#include <QtWidgets>
#include <string>
#include <typeinfo>
#include <iostream>
#include "../core/family_tree.h"
#include "../core/strings.h"
#include "mainwindow.h"

```

## Classes

- class [DateDialog](#)  
*Dialog for editing custom dates.*
- class [TagChooserDialog](#)  
*Dialog for editing or creating tags for a person.*
- class [AddFileDialog](#)  
*Dialog for adding file.*
- class [RelationTemplateDialog](#)  
*Dialog showing all templates and possibly editing them or creating new ones.*
- class [EventTemplatesDialog](#)  
*Dialog showing all event templates and possibly editing them or creating new ones.*
- class [ChoosePersonDialog](#)  
*Dialog for choosing person with find window. Parent have to have slot `savePerson()`.*
- class [RelationDialog](#)  
*Dialog showing single relation for editing it.*
- class [EventDialog](#)  
*Dialog showing single event for editing it.*
- class [HelpDialog](#)  
*Dialog with text edit to show help or other texts.*
- class [VisualSettings](#)  
*Dialog for editing and showing graphic settings.*
- class [ExportTemplatesDialog](#)  
*Dialog for exporting some of the event and relation templates.*
- class [ExportProjectDialog](#)  
*Dialog for exporting project.*
- class [PictureExport](#)  
*Dialog for exporting diagram. To choose zoom of the picture.*
- class [HtmlExport](#)  
*Dialog for exporting HTML for one person and all persons.*
- class [SuggestionsDialog](#)  
*Dialog for relation suggestions.*

## Namespaces

- namespace [Ui](#)  
*Namespace for classes having all its user interface items.*

### 15.22.1 Detailed Description

This file contains all window dialogs that will be shown. These classes are relatively small but there are many of them.

## 15.23 dialogs.h

[Go to the documentation of this file.](#)

```

00001
00002 #ifndef dialogs_h_
00003 #define dialogs_h_
00004
00005 // Header files created from .ui files.
00006 #include "ui_date_dialog.h"
00007 #include "ui_tag_chooser.h"
00008 #include "ui_add_file.h"
00009 #include "ui_relation_templates.h"
00010 #include "ui_event_templates.h"
00011 #include "ui_event_dialog.h"
00012 #include "ui_relation_dialog.h"
00013 #include "ui_help_dialog.h"
00014 #include "ui_choose_person.h"
00015 #include "ui_visual_settings.h"
00016 #include "ui_export_templates.h"
00017 #include "ui_export_project.h"
00018 #include "ui_html_export.h"
00019 #include "ui_picture_export.h"
00020 #include "ui_suggestions.h"
00021
00022 #include "graphics_items.h"
00023 #include <QtCore>
00024 #include <QtWidgets>
00025 #include <string>
00026 #include <typeinfo>
00027 #include <iostream>
00028 #include "../core/family_tree.h"
00029 #include "../core/strings.h"
00030 #include "mainwindow.h"
00031
00032 QT_BEGIN_NAMESPACE
00033 namespace Ui {
00034     class DateDialog;
00035     class TagChooserDialog;
00036     class AddFileDialog;
00037     class RelationTemplateDialog;
00038     class EventTemplatesDialog;
00039     class EventDialog;
00040     class RelationDialog;
00041     class HelpDialog;
00042     class ChoosePersonDialog;
00043     class VisualSettings;
00044     class ExportTemplatesDialog;
00045     class ExportProjectDialog;
00046     class PictureExport;
00047     class HtmlExport;
00048     class SuggestionsDialog;
00049 }
00050 QT_END_NAMESPACE
00051
00052 class DateDialog : public QDialog {
00053     Q_OBJECT
00054     public:
00055         explicit DateDialog(FamilyTree* FT, WrappedDate* date = nullptr, QWidget* parent = nullptr);
00056         ~DateDialog();
00057     public slots:
00058         void saveDate();
00059         void unknownDate(bool isUnknown);
00060     private:
00061         WrappedDate* date_;
00062         FamilyTree* FT_;
00063         Ui::DateDialog *ui;
00064 };
00065
00066 class TagChooserDialog : public QDialog{
00067     Q_OBJECT
00068     public:
00069         TagChooserDialog(FamilyTree* FT, const QString& tag, const QString& value, QWidget* parent =
00070             nullptr);
00071         ~TagChooserDialog();
00072     public slots:
00073         void changeVisibleWidgets(int index);
00074         void saveTag();
00075     private:
00076         FamilyTree* FT_;
00077         bool newOne_;
00078         Ui::TagChooserDialog *ui;
00079 };
00080
00081 class AddFileDialog : public QDialog{
00082     Q_OBJECT

```

```

00124     public:
00131         AddFileDialog(FileType type, const std::vector<std::string>& extensions, VirtualDrive*
parentDrive, FamilyTree* FT, QWidget* parent);
00133     ~AddFileDialog();
00134     public slots:
00136         void copyFile();
00138         void createOrChooseFile();
00141         void openFile(QTreeWidgetItem* item);
00142     private:
00144         FamilyTree* FT_;
00148         size_t getId(QTreeWidgetItem* item);
00150         VirtualDrive* drive_;
00152         FileType type_;
00154         Ui::AddFileDialog* ui;
00155 };
00156
00158 class RelationTemplateDialog : public QDialog{
00159     Q_OBJECT
00160     public:
00164         explicit RelationTemplateDialog(FamilyTree* FT, QWidget* parent = nullptr);
00166         ~RelationTemplateDialog();
00167     public slots:
00169         void addDefaults();
00171         void addNewTemplate();
00173         void removeTemplate();
00175         void saveTemplate();
00178         void showTemplate(QTreeWidgetItem* item);
00179     private:
00181         bool addingNew_;
00183         void clear();
00185         FamilyTree* FT_;
00187         void refreshListWidget();
00190         void setEnabledWidgets(bool enabled);
00193         void saveTemplate(RelationTemplate* rel);
00195         Ui::RelationTemplateDialog* ui;
00196 };
00197
00199 class EventTamplesDialog : public QDialog{
00200     Q_OBJECT
00201     public:
00205         explicit EventTamplesDialog(FamilyTree* FT, QWidget* parent = nullptr);
00207         ~EventTamplesDialog();
00208     public slots:
00210         void addDefaults();
00212         void addNewRole();
00214         void addNewTemplate();
00216         void removeRole();
00218         void removeTemplate();
00220         void saveTemplate();
00223         void showTemplate(QTreeWidgetItem* item);
00224     private:
00226         bool addingNew_;
00228         void clear();
00230         FamilyTree* FT_;
00232         void refreshListWidget();
00235         void saveTemplate(EventTemplate* event);
00238         void setEnabledWidgets(bool enabled);
00240         Ui::EventTamplesDialog* ui;
00241 };
00242
00244 class ChoosePersonDialog : public QDialog{
00245     Q_OBJECT
00246     public:
00251         ChoosePersonDialog(FamilyTree* FT, Person* person, QWidget* parent = nullptr);
00253         ~ChoosePersonDialog();
00254     signals:
00256         void savePerson(Person* person);
00257     public slots:
00259         void clearFilter();
00261         void filterPersons();
00263         void saveSelectedPerson();
00264     private:
00266         FamilyTree* FT_;
00268         Person* person_;
00270         Ui::ChoosePersonDialog* ui;
00271 };
00272
00274 class RelationDialog : public QDialog{
00275     Q_OBJECT
00276     public:
00281         explicit RelationDialog(FamilyTree* FT, QWidget* parent = nullptr, Relation* rel = nullptr);
00283         ~RelationDialog();
00284     public slots:
00286         void chooseFirstPerson();
00288         void chooseSecondPerson();
00290         void savePerson(Person* person);
00292         void saveRelation();

```

```

00294     void showTemplateDetails();
00296     void switchPersons();
00297 private:
00299     FamilyTree* FT_;
00301     Person* person1_;
00303     Person* person2_;
00305     Relation* rel_;
00307     bool saveFirstPerson_;
00309     Ui::RelationDialog* ui;
00310 };
00311
00313 class EventDialog : public QDialog{
00314     Q_OBJECT
00315 public:
00320     explicit EventDialog(FamilyTree* FT, QWidget* parent = nullptr, Event* event = nullptr);
00322     ~EventDialog();
00323 public slots:
00325     void addEmptyRole();
00327     void choosePerson();
00329     void openCustomDate();
00331     void refreshDates();
00334     void refreshPersons(QTreeWidgetItem* item);
00336     void removeEmptyRole();
00338     void removePerson();
00340     void saveEvent();
00343     void savePerson(Person* person);
00345     void showTemplate();
00346 private:
00348     WrappedDate date_;
00350     Event* event_;
00352     FamilyTree* FT_;
00354     Person* main_;
00356     void showEventDetails();
00358     Ui::EventDialog* ui;
00359 };
00360 };
00361
00363 class HelpDialog : public QDialog{
00364     Q_OBJECT
00365 public:
00369     explicit HelpDialog(FamilyTree* FT, QWidget* parent = nullptr);
00371     ~HelpDialog();
00372 public slots:
00375     void openHelp(bool help);
00376 private:
00378     FamilyTree* FT_;
00380     Ui::HelpDialog* ui;
00381 };
00382
00384 class VisualSettings : public QDialog{
00385     Q_OBJECT
00386 public:
00399     VisualSettings(QColor probandColor, QColor promotedColor, QColor standardColor, QColor
highlighted, int sizeX, int sizeY, QFont font,
00400                    QPen* pen, QPen* textPen, int borderRadius, QWidget* parent = nullptr);
00402     ~VisualSettings();
00403 signals:
00416     void saveSettings(QColor probandColor, QColor promotedColor, QColor standardColor, QColor
highlightedColor, int sizeX, int sizeY, QFont font,
00417                    QColor lineColor, QColor textColor, int lineWidth, int rounding);
00418 public slots:
00420     void highlightedColor();
00422     void lineColor();
00424     void refresh();
00426     void probandColor();
00428     void promotedColor();
00430     void saveAndExit();
00432     void standardColor();
00434     void textColor();
00435 private:
00437     int borderRadius_;
00439     QFont font_;
00441     QColor highlightedColor_;
00443     QPen pen_;
00445     QColor probandColor_;
00447     QColor promotedColor_;
00449     QGraphicsScene* scene;
00451     int sizeX_;
00453     int sizeY_;
00455     QColor standardColor_;
00457     QPen textPen_;
00459     Ui::VisualSettings* ui;
00460 };
00461
00463 class ExportTemplatesDialog : public QDialog{
00464     Q_OBJECT
00465 public:

```

```

00469     explicit ExportTemplatesDialog(FamilyTree* FT, QWidget* parent = nullptr);
00471     ~ExportTemplatesDialog();
00472     public slots:
00474         void exportTemplates();
00475     private:
00477         FamilyTree* FT_;
00479         Ui::ExportTemplatesDialog* ui;
00480 };
00481
00483 class ExportProjectDialog : public QDialog{
00484     Q_OBJECT
00485     public:
00489         explicit ExportProjectDialog(FamilyTree* FT, QWidget* parent = nullptr);
00491         ~ExportProjectDialog();
00492     public slots:
00494         void checkPerson();
00496         void exportProject();
00497     private:
00499         FamilyTree* FT_;
00501         Ui::ExportProjectDialog* ui;
00502 };
00503
00505 class PictureExport : public QDialog{
00506     Q_OBJECT
00507     public:
00511         explicit PictureExport(bool closeDiagram, QWidget* parent = nullptr);
00513         ~PictureExport();
00514     public slots:
00516         void exportDiagram();
00517     private:
00519         bool closeDiagram_;
00521         Ui::PictureExport* ui;
00522 };
00523
00524
00526 class HtmlExport : public QDialog{
00527     Q_OBJECT
00528     public:
00535         HtmlExport(FamilyTree* FT, bool all, int up, int down, QWidget* parent = nullptr);
00537         ~HtmlExport();
00538     public slots:
00541         void allAboveGenerations(bool all);
00544         void allBelowGenerations(bool all);
00546         void exportAllHtml();
00548         void exportMainHtml();
00549     private:
00551         FamilyTree* FT_;
00553         Ui::HtmlExport* ui;
00554 };
00555
00557 class SuggestionsDialog : public QDialog{
00558     Q_OBJECT
00559     public:
00564         SuggestionsDialog(FamilyTree* FT, std::vector<RelationSuggestion> suggestions, QWidget* parent
= nullptr);
00566         ~SuggestionsDialog();
00567     public slots:
00569         void removeSuggestion();
00571         void save();
00574         void selectedSuggestion(QTreeWidgetItem* item);
00576         void swapPersons();
00578         void switchedTemplate();
00579     private:
00581         std::vector<RelationTemplate*> fatherhoods;
00583         FamilyTree* FT_;
00585         std::vector<RelationTemplate*> motherhoods;
00587         std::vector<RelationTemplate*> partnerships;
00589         std::vector<RelationTemplate*> siblings;
00591         Ui::SuggestionsDialog* ui;
00592 };
00593
00594 #endif

```

## 15.24 graphics/graphics\_items.cpp File Reference

This files contains implementation of all graphics items, that are inherited from Qt framework.

```
#include "graphics_items.h"
```

## Functions

- bool [containsForb](#) (const QString &value)

### 15.24.1 Detailed Description

This files contains implementation of all graphics items, that are inherited from Qt framework.

### 15.24.2 Function Documentation

#### 15.24.2.1 containsForb()

```
bool containsForb (
    const QString & value )
```

To clarify that the value contains a forbidden character.

#### Parameters

|              |                         |
|--------------|-------------------------|
| <i>value</i> | Which value is checked. |
|--------------|-------------------------|

#### Returns

True if it contains any of forbidden characters.

## 15.25 graphics/graphics\_items.h File Reference

This files contains all graphics items, that are inherited from Qt framework.

```
#include <QtCore>
#include <QtWidgets>
#include <string>
#include <typeinfo>
#include <iostream>
#include "../core/family_tree.h"
#include "../core/strings.h"
```

## Classes

- class [FileTreeItem](#)  
*TreeWidget Item for general file.*
- class [ProjectItem](#)  
*This is a tree widget item to be shown on a project view.*
- class [EventRoleItem](#)  
*Tree widget item for persons in events.*
- class [PersonsGraphicsItem](#)



- Box with persons information.*
- class [CellGraphicsItem](#)  
*Cell around all relations.*
- class [GraphicsView](#)  
*Class for creating QGraphicsView with using wheel for zoom in and out. And grabing by mouse moves the view.*
- class [RelationItem](#)  
*Class for showing relation suggestion and holding the data.*

## Enumerations

- enum [ProjectItemType](#) {  
[PROJECT\\_PERSON](#) , [PROJECT\\_GROUP](#) , [PROJECT\\_FILE](#) , [PROJECT\\_MEDIA](#) ,  
[PROJECT\\_NOTE](#) , [PROJECT\\_EVENTS](#) , [PROJECT\\_RELATIONS](#) }  
*What type of a item is to be shown.*

## Functions

- bool [containsForb](#) (const QString &value)

### 15.25.1 Detailed Description

This files contains all graphics items, that are inherited from Qt framework.

### 15.25.2 Enumeration Type Documentation

#### 15.25.2.1 ProjectItemType

```
enum ProjectItemType
```

What type of a item is to be shown.

#### Enumerator

|                   |  |
|-------------------|--|
| PROJECT_PERSON    |  |
| PROJECT_GROUP     |  |
| PROJECT_FILE      |  |
| PROJECT_MEDIA     |  |
| PROJECT_NOTE      |  |
| PROJECT_EVENTS    |  |
| PROJECT_RELATIONS |  |

### 15.25.3 Function Documentation

#### 15.25.3.1 containsForb()

```
bool containsForb (
    const QString & value )
```

To clarify that the value contains a forbidden character.

#### Parameters

| <i>value</i> | Which value is checked. |
|--------------|-------------------------|
|--------------|-------------------------|

#### Returns

True if it contains any of forbidden characters.

## 15.26 graphics\_items.h

[Go to the documentation of this file.](#)

```

00001
00002 #ifndef graphics_items_h_
00003 #define graphics_items_h_
00004
00005 #include <QtCore>
00006 #include <QtWidgets>
00007 #include <string>
00008 #include <typeinfo>
00009 #include <iostream>
00010 #include "../core/family_tree.h"
00011 #include "../core/strings.h"
00012
00016 bool containsForb(const QString& value);
00017
00019 class FileTreeItem : public QTreeWidgetItem{
00020     public:
00026         FileTreeItem(VirtualDrive* drive, FamilyTree* FT, const QIcon& icon, FileType type);
00032         FileTreeItem(size_t id, FamilyTree* FT, const QIcon& icon, FileType type);
00036         void addFile(size_t id, const QIcon& icon);
00039         void addSubDrive(const QString& folder);
00042         VirtualDrive* getDrive();
00045         size_t getId();
00048         bool isDirectory();
00051         bool removeFolder();
00055         bool rename(const QString& name);
00056     private:
00058         VirtualDrive* drive_;
00060         size_t fileId_;
00062         FamilyTree* FT_;
00064         bool isDir_;
00067         void removeFile(size_t id);
00069         FileType type_;
00070 };
00071
00073 enum ProjectItemType {PROJECT_PERSON, PROJECT_GROUP, PROJECT_FILE, PROJECT_MEDIA, PROJECT_NOTE,
PROJECT_EVENTS, PROJECT_RELATIONS};
00074
00076 class ProjectItem : public QTreeWidgetItem{
00077     public:
00081         explicit ProjectItem(ProjectItemType type, size_t id = 0);
00084         size_t getId() const;
00087         ProjectItemType type() const;
00088     private:
00090         size_t id_;
00092         ProjectItemType type_;
00093 };
00094
00096 class EventRoleItem : public QTreeWidgetItem{
00097     public:
00101         explicit EventRoleItem(const std::string& role, size_t id = 0);
00104         size_t getId();
00108         void setPerson(const QString& name, size_t id = 0);
00109     private:
00111         size_t id_;
00112 };
00113
00115 class PersonsGraphicsItem : public QObject, public QGraphicsItem{
00116     Q_OBJECT
00117     public:
00132         PersonsGraphicsItem(const QStringList& lines, int sizex, int sizey, Person* p, qreal x, qreal
y,
00133                             QColor color, QColor highlighted, QFont font, QWidget* parent, QPen* pen,
QPen* textPen, int borderRadius);

```

```

00135     ~PersonsGraphicsItem();
00138     QRectF boundingRect() const override;
00143     void paint(QPainter* painter, const QStyleOptionGraphicsItem* option, QWidget* widget)
00144     override;
00146     QPainterPath shape() const override;
00147     signals:
00149     void itemClicked(size_t personId);
00150     protected:
00153     void hoverEnterEvent(QGraphicsSceneHoverEvent *event) override;
00156     void hoverLeaveEvent(QGraphicsSceneHoverEvent *event) override;
00159     void mousePressEvent(QGraphicsSceneMouseEvent* event) override;
00160     private:
00162     int borderRadius_;
00164     int bold_text_;
00166     QColor color_;
00168     QFont font_;
00170     QColor highlighted_;
00172     bool hovered_;
00174     QPen* pen_;
00176     Person* person_;
00178     int size_x_;
00180     int size_y_;
00182     QStringList textLines_;
00184     QPen* textPen_;
00186     qreal x_;
00188     qreal y_;
00189 };
00190
00192 class CellGraphicsItem : public QGraphicsItem{
00193     public:
00201     CellGraphicsItem(int size_x, int size_y, qreal x, qreal y, QPen* pen, int borderRadius);
00204     QRectF boundingRect() const override;
00209     void paint(QPainter* painter, const QStyleOptionGraphicsItem* option, QWidget* widget)
00210     override;
00212     QPainterPath shape() const override;
00213     private:
00215     int borderRadius_;
00217     QPen* pen_;
00219     int size_x_;
00221     int size_y_;
00223     qreal x_;
00225     qreal y_;
00226 };
00227
00229 class GraphicsView : public QGraphicsView {
00230     public:
00234     explicit GraphicsView(QGraphicsScene* scene, QWidget* parent = nullptr);
00235     protected:
00238     void wheelEvent(QWheelEvent* event) override;
00239 };
00240
00242 class RelationItem : public QTreeWidgetItem{
00243     public:
00248     RelationItem(FamilyTree* FT, const RelationSuggestion& suggestion, RelationTemplate* templ);
00251     Relation* getRelation();
00253     void refreshTexts();
00254     private:
00256     FamilyTree* FT_;
00258     Relation relation_;
00259 };
00260
00261 #endif

```

## 15.27 graphics/mainwindow.cpp File Reference

Implementation of all standard methods. For slots see [mainwindow\\_slots.cpp](#).

```
#include "mainwindow.h"
```

### 15.27.1 Detailed Description

Implementation of all standard methods. For slots see [mainwindow\\_slots.cpp](#).

## 15.28 graphics/mainwindow.h File Reference

This header file contains everything around the main window.

```
#include <QtCore>
#include <QtWidgets>
#include <string>
#include <cstdlib>
#include "ui_mainwindow.h"
#include "dialogs.h"
#include "graphics_items.h"
#include "../core/family_tree.h"
#include "../core/strings.h"
```

### Classes

- class [MainWindow](#)

*Class representing main window using Qt library.*

### Namespaces

- namespace [Ui](#)

*Namespace for classes having all its user interface items.*

### 15.28.1 Detailed Description

This header file contains everything around the main window.

## 15.29 mainwindow.h

[Go to the documentation of this file.](#)

```
00001
00002 #ifndef mainwindow_h_
00003 #define mainwindow_h_
00004
00005 #include <QtCore>
00006 #include <QtWidgets>
00007 #include <string>
00008 #include <cstdlib>
00009
00010 #include "ui_mainwindow.h"
00011
00012 #include "dialogs.h"
00013 #include "graphics_items.h"
00014 #include "../core/family_tree.h"
00015 #include "../core/strings.h"
00016
00017 QT_BEGIN_NAMESPACE
00018 namespace Ui {
00019     class MainWindow;
00020 }
00021 QT_END_NAMESPACE
00022
00023 class MainWindow : public QMainWindow {
00024     Q_OBJECT
00025 public:
00026     explicit MainWindow(QWidget *parent = nullptr);
00027     ~MainWindow();
```

```

00038     bool exportCloseFamilyDiagram(double scale, const std::string& filename = EMPTY_STRING);
00043     bool exportTreeDiagram(double scale, const std::string& filename = EMPTY_STRING);
00048     std::pair<int, int> setNumberOfGenerations(int up, int down);
00051     static void openExternalFile(const std::string& fileURL);
00052 public slots:
00054     void addNewEvent();
00056     void addNewFile();
00058     void addNewFileFolder();
00060     void addNewMedia();
00062     void addNewMediaFolder();
00064     void addNewNote();
00066     void addNewNoteFolder();
00068     void addNewPerson();
00070     void addNewRelation();
00073     void allAboveGenerations(bool selected);
00076     void allBelowGenerations(bool selected);
00079     void changeGenerationsDown(int value);
00082     void changeGenerationsUp(int value);
00084     void checkStoreRefreshPersonsInfo();
00086     void clearProjectFilter();
00088     void clearProjectPaths();
00090     void disableFileFolderPrompt();
00092     void disableMediaFolderPrompt();
00094     void disableNoteFolderPrompt();
00097     void editEvent(QTreeWidgetItem* item);
00100     void editRelation(QTreeWidgetItem* item);
00103     void editTag(QTreeWidgetItem* item = nullptr);
00105     void enableFileFolderPrompt();
00107     void enableMediaFolderPrompt();
00109     void enableNoteFolderPrompt();
00111     void exportCloseFamilyDiagramDialog();
00113     void exportHtml();
00115     void exportMainHtml();
00117     void exportProject();
00119     void exportTemplates();
00121     void exportTreeDiagramDialog();
00123     void filterProjectItems();
00125     void findPerson();
00127     void fitTreeInView();
00129     void importProject();
00131     void importRootDir();
00133     void importTemplates();
00138     void interactWithProjectItem(QTreeWidgetItem* item);
00140     void openBirthDateDialog();
00142     void openDeathDateDialog();
00144     void openEventTemplatesDialog();
00146     void openExistingProject();
00148     void openGraphicSettings();
00150     void openNewFamilyTreeProject();
00152     void openRelationTemplatesDialog();
00155     void openSelectedFile(QTreeWidgetItem* item);
00158     void openSelectedMedia(QTreeWidgetItem* item);
00161     void openSelectedNote(QTreeWidgetItem* item);
00163     void promoteRelation();
00165     void refreshDates();
00167     void refreshEventTab();
00169     void refreshGraphics();
00171     void refreshProjectView();
00173     void refreshRelationTab();
00175     void refreshTabsWithFiles();
00177     void refreshTagsList();
00179     void removeEvent();
00181     void removeFileFolder();
00183     void removeMediaFolder();
00185     void removeNoteFolder();
00187     void removePerson();
00189     void removeRelation();
00191     void removeTag();
00193     void renameFile();
00195     void renameMedia();
00197     void renameNote();
00200     bool saveDatabase();
00213     void saveGraphicsSetting(QColor probandColor, QColor promotedColor, QColor standardColor, QColor
highlightedColor, int sizeX, int sizeY, QFont font,
                                QColor lineColor, QColor textColor, int lineWidth, int rounding);
00214     void selectOlderGenerationOnly();
00218     void selectSameGenerationOnly();
00220     void selectYoungerGenerationOnly();
00223     void setEnabledDeathAttributes(bool disabled);
00225     void setEventTabVisible();
00227     void setFileTabVisible();
00229     void setGeneralTabVisible();
00232     void setMainPerson(size_t personId);
00234     void setMediaTabVisible();
00236     void setNoteTabVisible();
00238     void setRelationTabVisible();
00240     void setTabEast();

```

```

00242     void setTabNorth();
00244     void setTabSouth();
00246     void setTabWest();
00248     void showHelp();
00250     void showRenameFile();
00252     void showRenameMedia();
00254     void showRenameNote();
00256     void splitterChanged();
00258     void switchNewTreeProject();
00260     void zoomIn();
00262     void zoomOut();
00263 protected:
00266     void closeEvent(QCloseEvent *event);
00267 private:
00272     void addGeneralFile(FileType type, const std::vector<std::string>& extensions, FileTreeItem*
item);
00275     void addGeneralFolder(FileType type);
00277     int borderRadius_;
00279     void clearUi();
00281     void connectAllSlots();
00284     void editEvent(size_t id);
00287     void editRelation(size_t id);
00291     void enableGeneralFolderPrompt(bool visible, FileType type);
00294     void enableGeneralFolderRenamePrompt(FileType type);
00300     bool exportGeneralDiagram(const QString& filename, QGraphicsScene* gscene, double scale = 1);
00302     QFont font_;
00304     FamilyTree FT;
00331     void drawCloseContainer(const std::vector<Relation*>& container,
00332                             qreal sizeX, qreal sizeY, qreal topLeftX, qreal topLeftY,
00333                             qreal lineX1, qreal lineY1, qreal lineX2, qreal lineY2,
00334                             size_t partner, size_t mother, size_t father, int gap,
00335                             bool movex, bool movey);
00350     void drawCloseFamily();
00369     void drawFamilyTree();
00382     void drawGeneralGenerations(Person* p, qreal width, qreal midx, qreal midy, int horizontalGap, int
verticalGap, size_t barrier,
00383                                     const std::vector<size_t>& container, bool up,
00384                                     std::function<std::pair<size_t, size_t>(Person*, size_t, size_t)>
numberFunc,
00385                                     std::function<void(Person*, qreal, qreal, qreal, int, int,
size_t)> drawFunc);
00401     void drawOlderGenerations(Person* p, qreal width, qreal midx, qreal midy, int horizontalGap, int
verticalGap, size_t barrier);
00406     void drawPartner(qreal canvasMidX, qreal canvasMidY, int horizontalGap);
00424     void drawYoungerGenerations(Person* p, qreal width, qreal midx, qreal midy, int horizontalGap, int
verticalGap, size_t barrier);
00433     std::pair<size_t, size_t> generalGenerationNumber(Person* p, size_t soFar, size_t barrier,
00434                                                         const std::vector<size_t>& container, bool up,
00435                                                         std::function<std::pair<size_t, size_t>(Person*,
size_t, size_t)> func);
00437     int genSizeDown_;
00439     int genSizeUp_;
00441     QColor highlightedColor_;
00443     QPen linePen_;
00445     void loadSettings();
00451     std::pair<size_t, size_t> olderGenerationNumber(Person* p, int soFar, size_t barrier);
00454     void openCustomDateDialog(WrappedDate* date);
00458     void initializeProjectSubView(ProjectItem* parent, Person* person);
00460     void initializeProjectView();
00464     void openGeneralFile(QTreeWidgetItem* item, FileType type);
00467     void openGeneralProject(const std::string& dirPath);
00469     QColor probandColor_;
00471     void refreshPersonsInfoTab();
00475     void refreshProjectEventsView(ProjectItem* item, Person* person);
00483     void refreshProjectGeneralFilesView(const QIcon& containerIcon, const QString& containerName,
const QIcon& fileIcon, FileType type, ProjectItem* item, Person* person);
00487     void refreshProjectFilesView(ProjectItem* item, Person* person);
00491     void refreshProjectMediaView(ProjectItem* item, Person* person);
00495     void refreshProjectNoteView(ProjectItem* item, Person* person);
00499     void refreshProjectRelationsView(ProjectItem* item, Person* person);
00501     void refreshUi();
00504     void removeGeneralFolder(FileType type);
00507     void renameGeneralFileItem(FileType type);
00509     QGraphicsScene* scene;
00511     void savePersonsInfo();
00513     void setUiElements();
00518     void showGeneralFiles(QTreeWidgetItem* widget, VirtualDrive* root, FileType type);
00520     int sizeX_;
00522     int sizeY_;
00524     QColor promotedColor_;
00526     QColor standardColor_;
00528     QPen textPen_;
00530     QGraphicsScene* treeScene;
00532     GraphicsView* treeView;
00534     Ui::MainWindow* ui;
00540     std::pair<size_t, size_t> youngerGenerationNumber(Person* p, size_t soFar, size_t barrier);
00541 };

```

```
00542
00543 #endif
```

## 15.30 graphics/mainwindow\_slots.cpp File Reference

Implementation of all slots of the main window. For other methods see [mainwindow.cpp](#).

```
#include "mainwindow.h"
```

### 15.30.1 Detailed Description

Implementation of all slots of the main window. For other methods see [mainwindow.cpp](#).

## 15.31 graphics/resources/icons.md File Reference

## 15.32 help texts/events.md File Reference

## 15.33 help texts/files.md File Reference

## 15.34 help texts/generic.md File Reference

## 15.35 help texts/layout.md File Reference

## 15.36 help texts/persons.md File Reference

## 15.37 help texts/relations.md File Reference

## 15.38 index.md File Reference

## 15.39 main.cpp File Reference

The main file for starting the application.

```
#include "graphics/mainwindow.h"
#include "core/strings.h"
#include <iostream>
```

## Functions

- int `main` (int argc, char \*\*argv)

### 15.39.1 Detailed Description

The main file for starting the application.

### 15.39.2 Function Documentation

#### 15.39.2.1 `main()`

```
int main (  
    int argc,  
    char ** argv )
```

Main function for the application.

#### Parameters

|             |                        |
|-------------|------------------------|
| <i>argc</i> | Number of arguments.   |
| <i>argv</i> | List of all arguments. |

#### Returns

The exit code of the application.



# Index

- ~AddFileDialog
  - AddFileDialog, [62](#)
- ~ChoosePersonDialog
  - ChoosePersonDialog, [73](#)
- ~DateDialog
  - DateDialog, [81](#)
- ~EventDialog
  - EventDialog, [91](#)
- ~EventTemplatesDialog
  - EventTemplatesDialog, [96](#)
- ~ExportProjectDialog
  - ExportProjectDialog, [106](#)
- ~ExportTemplatesDialog
  - ExportTemplatesDialog, [108](#)
- ~HelpDialog
  - HelpDialog, [149](#)
- ~HtmlExport
  - HtmlExport, [151](#)
- ~MainWindow
  - MainWindow, [158](#)
- ~PersonsGraphicsItem
  - PersonsGraphicsItem, [218](#)
- ~PictureExport
  - PictureExport, [223](#)
- ~RelationDialog
  - RelationDialog, [233](#)
- ~RelationTemplateDialog
  - RelationTemplateDialog, [247](#)
- ~SuggestionsDialog
  - SuggestionsDialog, [260](#)
- ~TagChooserDialog
  - TagChooserDialog, [263](#)
- ~VisualSettings
  - VisualSettings, [272](#)
- ADD\_NEW
  - button, [26](#)
- addDefaultEvent
  - Settings, [250](#)
- addDefaultEvents
  - Settings, [251](#)
- addDefaultRelation
  - Settings, [251](#)
- addDefaultRelations
  - Settings, [251](#)
- addDefaults
  - EventTemplatesDialog, [97](#)
  - RelationTemplateDialog, [247](#)
- addEmptyRole
  - EventDialog, [91](#)
- addEvent
  - FamilyTree, [112](#)
  - Person, [199](#)
- addEventTemplate
  - Settings, [251](#)
- addFile
  - FamilyTree, [112](#)
  - FileTreeItem, [144](#)
  - VirtualDrive, [265](#)
- addfile, [25](#)
  - NONE, [25](#)
- AddFileDialog, [61](#)
  - ~AddFileDialog, [62](#)
  - AddFileDialog, [62](#)
  - copyFile, [62](#)
  - createOrChooseFile, [62](#)
  - drive\_, [63](#)
  - FT\_, [63](#)
  - getId, [63](#)
  - openFile, [63](#)
  - type\_, [63](#)
  - ui, [64](#)
- addGeneralFile
  - MainWindow, [159](#)
- addGeneralFolder
  - MainWindow, [159](#)
- addingNew\_
  - EventTemplatesDialog, [98](#)
  - RelationTemplateDialog, [248](#)
- addNewEvent
  - MainWindow, [159](#)
- addNewFile
  - MainWindow, [159](#)
- addNewFileFolder
  - MainWindow, [159](#)
- addNewMedia
  - MainWindow, [160](#)
- addNewMediaFolder
  - MainWindow, [160](#)
- addNewNote
  - MainWindow, [160](#)
- addNewNoteFolder
  - MainWindow, [160](#)
- addNewPerson
  - MainWindow, [160](#)
- addNewRelation
  - MainWindow, [160](#)
- addNewRole
  - EventTemplatesDialog, [97](#)

- addNewTemplate
  - EventTemplatesDialog, 97
  - RelationTemplateDialog, 247
- addPerson
  - Event, 84
  - FamilyTree, 113
- addRelation
  - FamilyTree, 113
  - Person, 199
- addRelationTemplate
  - Settings, 251
- addSubDrive
  - FileTreeItem, 144
- addSubdrive
  - VirtualDrive, 265, 266
- addTag
  - Person, 200
  - Settings, 252
- ADOPTED\_CHILD
  - relation, 55
- ADOPTIVE\_FATHER
  - relation, 55
- ADOPTIVE\_FATHERHOOD
  - relation, 55
- ADOPTIVE\_MOTHER
  - relation, 55
- ADOPTIVE\_MOTHERHOOD
  - relation, 55
- AFTER\_TITLE
  - jsonlabel, 40
- ALL\_PERSONS
  - html, 34
- allAboveGenerations
  - HtmlExport, 151
  - MainWindow, 160
- allBelowGenerations
  - HtmlExport, 151
  - MainWindow, 161
- allEvents\_
  - FamilyTree, 135
- allFiles\_
  - FamilyTree, 135
- AllGenders
  - gender, 32
- allMedia\_
  - FamilyTree, 135
- allNotes\_
  - FamilyTree, 135
- allPersons\_
  - FamilyTree, 135
- allRelations\_
  - FamilyTree, 135
- AllTraits
  - trait, 60
- APP\_SETTINGS
  - jsonlabel, 40
- append
  - FamilyTree, 114
- AppSettings, 64
  - fontFamily, 66
  - fontSize, 66
  - genSizeDown, 66
  - genSizeUp, 66
  - highlightedColor, 66
  - lineColor, 66
  - lineWidth, 66
  - probandColor, 67
  - promoted, 67
  - radius, 67
  - readJson, 65
  - setDefault, 65
  - showEventTab, 67
  - showFileTab, 67
  - showGeneralTab, 67
  - showMediaTab, 67
  - showNoteTab, 67
  - showRelationTab, 68
  - sizeX, 68
  - sizeY, 68
  - splitterPositionOne, 68
  - splitterPositionTwo, 68
  - standard, 68
  - tabPosition, 68
  - textColor, 68
- appSettings\_
  - Settings, 257
- BEST\_MAN
  - events, 29
- BIRTH
  - jsonlabel, 40
- BIRTH\_DATE
  - html, 34
- BIRTH\_PLACE
  - html, 34
- birthDate
  - Person, 200
- BLUE
  - jsonlabel, 40
- blue
  - Color, 76
- bold\_text\_
  - PersonsGraphicsItem, 220
- BORDER\_RADIUS
  - jsonlabel, 40
- borderRadius\_
  - CellGraphicsItem, 71
  - MainWindow, 185
  - PersonsGraphicsItem, 220
  - VisualSettings, 274
- boundingRect
  - CellGraphicsItem, 70
  - PersonsGraphicsItem, 218
- BRIDE
  - events, 29
- BROTHER
  - relation, 55

BROTHERHHOD  
    relation, 56  
button, 25  
    ADD\_NEW, 26  
    CANCEL, 26  
    REMOVE, 26  
CANCEL  
    button, 26  
CellGraphicsItem, 69  
    borderRadius\_, 71  
    boundingRect, 70  
    CellGraphicsItem, 70  
    paint, 70  
    pen\_, 71  
    shape, 71  
    size\_, 71  
    sizey\_, 71  
    x\_, 71  
    y\_, 71  
changeGenerationsDown  
    MainWindow, 161  
changeGenerationsUp  
    MainWindow, 161  
changeVisibleWidgets  
    TagChooserDialog, 263  
checkFileConsistence  
    FamilyTree, 114  
checkFileConsistency  
    Parser, 189  
checkFileTypeConsistence  
    FamilyTree, 114  
checkPerson  
    ExportProjectDialog, 106  
checkStoreRefreshPersonsInfo  
    MainWindow, 161  
CHILD  
    relation, 56  
chooseFirstPerson  
    RelationDialog, 233  
choosePerson  
    EventDialog, 91  
ChoosePersonDialog, 72  
    ~ChoosePersonDialog, 73  
    ChoosePersonDialog, 73  
    clearFilter, 73  
    filterPersons, 73  
    FT\_, 74  
    person\_, 74  
    savePerson, 74  
    saveSelectedPerson, 74  
    ui, 74  
chooseSecondPerson  
    RelationDialog, 233  
clear  
    EventTemplatesDialog, 97  
    FamilyTree, 115  
    Parser, 189  
    RelationTemplateDialog, 247  
    Settings, 252  
    WrappedDate, 276  
clearFilter  
    ChoosePersonDialog, 73  
clearPersons  
    Event, 84  
clearProjectFilter  
    MainWindow, 161  
clearProjectPaths  
    FamilyTree, 115  
    MainWindow, 162  
clearUi  
    MainWindow, 162  
CLOSE\_FAMILY  
    html, 34  
CLOSE\_FAMILY\_DIAGRAM  
    html, 34  
closeDiagram\_  
    PictureExport, 223  
closeEvent  
    MainWindow, 162  
Color, 75  
    blue, 76  
    Color, 75  
    green, 76  
    readJson, 76  
    red, 76  
color\_  
    PersonsGraphicsItem, 220  
config.cpp  
    operator<<, 281–283  
config.h  
    East, 285  
    Fatherhood, 285  
    Motherhood, 285  
    None, 285  
    North, 285  
    operator<<, 285, 286  
    Partnership, 285  
    Sibling, 285  
    South, 285  
    TabPosition, 284  
    Trait, 285  
    West, 285  
configPath\_  
    Parser, 196  
connectAllSlots  
    MainWindow, 162  
CONTAINS\_DATE  
    jsonlabel, 41  
CONTAINS\_PLACE  
    jsonlabel, 41  
containsBackup  
    Parser, 190  
containsBackupFile  
    Parser, 190  
containsDate  
    EventTemplate, 100

- containsDate\_
  - EventTemplate, 103
- containsForb
  - graphics\_items.cpp, 324
  - graphics\_items.h, 325
- containsPlace
  - EventTemplate, 100
- containsPlace\_
  - EventTemplate, 103
- copyDrive
  - VirtualDrive, 266
- copyFile
  - AddFileDialog, 62
  - FamilyTree, 115
  - Parser, 190
- core/config.cpp, 281
- core/config.h, 283, 287
- core/date.cpp, 289
- core/date.h, 290, 291
- core/family\_tree.cpp, 292
- core/family\_tree.h, 292, 293
- core/family\_tree\_items.cpp, 295
- core/family\_tree\_items.h, 296, 298
- core/file\_parser.cpp, 299
- core/file\_parser.h, 300, 301
- core/person.cpp, 302
- core/person.h, 303, 306
- core/strings.h, 307, 315
- CORRUPTED\_DATABASE
  - error, 27
- createAllDirs
  - Parser, 191
- createDatabase
  - FamilyTree, 115
- createEmptyFile
  - FamilyTree, 115
  - Parser, 191
- createJsonFile
  - Parser, 191
- createOrChooseFile
  - AddFileDialog, 62
- createResourcesDir
  - FamilyTree, 116
  - Parser, 191
- CUSTOM\_CSS
  - html, 34
- CUSTOM\_TAG
  - tagchooser, 59
- CUSTOM\_TAGS
  - html, 34
- DATE
  - html, 34
  - jsonlabel, 41
- Date, 76
  - Date, 77
  - day, 80
  - getDay, 78
  - getMonth, 78
  - getYear, 78
  - isEmpty, 78
  - isExact, 78
  - month, 80
  - read, 78
  - setDay, 79
  - setMonth, 79
  - setYear, 79
  - str, 79
  - year, 80
- date
  - Event, 84
- date.cpp
  - operator<<, 289, 290
- date.h
  - operator<<, 291
- date1\_
  - WrappedDate, 279
- date2\_
  - WrappedDate, 279
- date\_
  - DateDialog, 82
  - Event, 88
  - EventDialog, 93
- DATE\_FROM
  - person, 53
- DATE\_TO
  - person, 53
- DateDialog, 80
  - ~DateDialog, 81
  - date\_, 82
  - DateDialog, 81
  - FT\_, 82
  - saveDate, 81
  - ui, 82
  - unknownDate, 81
- dateOfBirth\_
  - Person, 213
- dateOfDeath\_
  - Person, 213
- day
  - Date, 80
- DEAD\_PERSON
  - events, 29
- DEATH
  - jsonlabel, 41
- DEATH\_DATE
  - html, 35
- DEATH\_PLACE
  - html, 35
- deathDate
  - Person, 200
- DESC\_ADOPTIVE\_FATHERHOOD
  - relation, 56
- DESC\_ADOPTIVE\_MOTHERHOOD
  - relation, 56
- DESC\_BROTHERHOOD
  - relation, 56

- DESC\_DIVORCE
  - events, [29](#)
- DESC\_FUNERAL
  - events, [29](#)
- DESC\_GRADUATION
  - events, [30](#)
- DESC\_SIBLINGS
  - relation, [56](#)
- DESC\_SISTERHOOD
  - relation, [56](#)
- DESC\_WEDDING
  - events, [30](#)
- DESCRIPTION
  - html, [35](#)
  - jsonlabel, [41](#)
- description\_
  - EventTemplate, [104](#)
  - RelationTemplate, [244](#)
- DIR
  - jsonlabel, [41](#)
- disableFileFolderPrompt
  - MainWindow, [162](#)
- disableMediaFolderPrompt
  - MainWindow, [162](#)
- disableNoteFolderPrompt
  - MainWindow, [162](#)
- DIVORCE
  - events, [30](#)
- drawCloseContainer
  - MainWindow, [163](#)
- drawCloseFamily
  - MainWindow, [163](#)
- drawFamilyTree
  - MainWindow, [164](#)
- drawGeneralGenerations
  - MainWindow, [164](#)
- drawOlderGenerations
  - MainWindow, [165](#)
- drawPartner
  - MainWindow, [165](#)
- drawYoungerGenerations
  - MainWindow, [166](#)
- drive\_
  - AddFileDialog, [63](#)
  - FileTreeItem, [146](#)
- DRIVES
  - jsonlabel, [41](#)
- East
  - config.h, [285](#)
- editEvent
  - MainWindow, [166](#)
- editRelation
  - MainWindow, [167](#)
- editTag
  - MainWindow, [167](#)
- EMPTY\_BOXES
  - error, [27](#)
- EMPTY\_STRING
  - strings.h, [315](#)
- enableFileFolderPrompt
  - MainWindow, [167](#)
- enableGeneralFolderPrompt
  - MainWindow, [167](#)
- enableGeneralFolderRenamePrompt
  - MainWindow, [168](#)
- enableHelpOnStartup
  - FamilyTree, [116](#)
- enableMediaFolderPrompt
  - MainWindow, [168](#)
- enableNoteFolderPrompt
  - MainWindow, [168](#)
- error, [26](#)
  - CORRUPTED\_DATABASE, [27](#)
  - EMPTY\_BOXES, [27](#)
  - EXCEPTION\_STR, [27](#)
  - FORBIDDEN\_CHARS, [27](#)
  - NO\_CHOSEN\_FILE, [27](#)
  - NO\_ROLE, [28](#)
  - UNKNOWN\_EXCEP\_STR, [28](#)
- ERROR\_LOG
  - parser, [51](#)
- EVE\_TEMPL
  - jsonlabel, [41](#)
- Event, [82](#)
  - addPerson, [84](#)
  - clearPersons, [84](#)
  - date, [84](#)
  - date\_, [88](#)
  - Event, [83](#)
  - getDate, [84](#)
  - getId, [84](#)
  - getPersons, [85](#)
  - getPlace, [85](#)
  - getSettings, [85](#)
  - getTemplate, [85](#)
  - getText, [85](#)
  - id\_, [88](#)
  - persons\_, [88](#)
  - place\_, [89](#)
  - printHtml, [86](#)
  - readJson, [86](#)
  - removePerson, [86](#)
  - setDate, [87](#)
  - setId, [87](#)
  - setPlace, [87](#)
  - setTemplate, [87](#)
  - setText, [88](#)
  - settings\_, [89](#)
  - str, [88](#)
  - template\_, [89](#)
  - text\_, [89](#)
  - updateToTemplate, [88](#)
- event\_
  - EventDialog, [93](#)
- event\_index\_
  - FamilyTree, [136](#)

- EventDialog, 89
  - ~EventDialog, 91
  - addEmptyRole, 91
  - choosePerson, 91
  - date\_, 93
  - event\_, 93
  - EventDialog, 91
  - FT\_, 93
  - main\_, 93
  - openCustomDate, 91
  - refreshDates, 91
  - refreshPersons, 91
  - removeEmptyRole, 92
  - removePerson, 92
  - saveEvent, 92
  - savePerson, 92
  - showEventDetails, 92
  - showTemplate, 92
  - ui, 93
- EventRoleItem, 93
  - EventRoleItem, 94
  - getId, 94
  - id\_, 95
  - setPerson, 94
- EVENTS
  - html, 35
  - jsonlabel, 42
- Events, 5
- events, 28
  - BEST\_MAN, 29
  - BRIDE, 29
  - DEAD\_PERSON, 29
  - DESC\_DIVORCE, 29
  - DESC\_FUNERAL, 29
  - DESC\_GRADUATION, 30
  - DESC\_WEDDING, 30
  - DIVORCE, 30
  - FORMER\_HUSBAND, 30
  - FORMER\_WIFE, 30
  - FUNERAL, 30
  - GRADUATE, 30
  - GRADUATION, 31
  - GROOM, 31
  - MAID\_OF\_HONOR, 31
  - NO\_ROLE, 31
  - WEDDING, 31
- events\_
  - Person, 213
- EventTemplatesDialog, 95
  - ~EventTemplatesDialog, 96
  - addDefaults, 97
  - addingNew\_, 98
  - addNewRole, 97
  - addNewTemplate, 97
  - clear, 97
  - EventTemplatesDialog, 96
  - FT\_, 98
  - refreshListWidget, 97
  - removeRole, 97
  - removeTemplate, 97
  - saveTemplate, 97
  - setEnabledWidgets, 98
  - showTemplate, 98
  - ui, 98
- EventTemplate, 99
  - containsDate, 100
  - containsDate\_, 103
  - containsPlace, 100
  - containsPlace\_, 103
  - description\_, 104
  - EventTemplate, 100
  - getDescription, 100
  - getId, 100
  - getRoles, 100
  - getTitle, 101
  - hasMorePeopleInvolved, 101
  - hasRoles, 101
  - id\_, 104
  - morePeopleInvolved\_, 104
  - readJson, 101
  - roles\_, 104
  - setContainsDate, 102
  - setContainsPlace, 102
  - setDescription, 102
  - setId, 102
  - setMorePeopleInvolved, 103
  - setRoles, 103
  - setTitle, 103
  - str, 103
  - title\_, 104
- eventTemplateIndex\_
  - Settings, 257
- eventTemplates\_
  - Settings, 258
- EXCEPTION\_STR
  - error, 27
- existsTag
  - Person, 200
- EXPORT
  - jsonlabel, 42
- exportAllHtml
  - HtmlExport, 151
- exportCloseFamilyDiagram
  - MainWindow, 168
- exportCloseFamilyDiagramDialog
  - MainWindow, 169
- exportDiagram
  - PictureExport, 223
- exportGeneralDiagram
  - MainWindow, 169
- exportHtml
  - MainWindow, 169
- exportJson
  - Person, 201
- exportMainHtml
  - HtmlExport, 151

- MainWindow, 169
- exportProject
  - ExportProjectDialog, 106
  - FamilyTree, 116
  - MainWindow, 169
- ExportProjectDialog, 105
  - ~ExportProjectDialog, 106
  - checkPerson, 106
  - exportProject, 106
  - ExportProjectDialog, 105
  - FT\_, 106
  - ui, 106
- exportTemplates
  - ExportTemplatesDialog, 108
  - FamilyTree, 117
  - MainWindow, 169
  - Settings, 252
- ExportTemplatesDialog, 107
  - ~ExportTemplatesDialog, 108
  - exportTemplates, 108
  - ExportTemplatesDialog, 107
  - FT\_, 108
  - ui, 108
- exportTreeDiagram
  - MainWindow, 170
- exportTreeDiagramDialog
  - MainWindow, 170
- FAMILY\_TREE
  - html, 35
- FAMILY\_TREE\_DIAGRAM
  - html, 35
- family\_tree\_items.cpp
  - operator<<, 295, 296
- family\_tree\_items.h
  - operator<<, 297
- FamilyTree, 108
  - addEvent, 112
  - addFile, 112
  - addPerson, 113
  - addRelation, 113
  - allEvents\_, 135
  - allFiles\_, 135
  - allMedia\_, 135
  - allNotes\_, 135
  - allPersons\_, 135
  - allRelations\_, 135
  - append, 114
  - checkFileConsistence, 114
  - checkFileTypeConsistence, 114
  - clear, 115
  - clearProjectPaths, 115
  - copyFile, 115
  - createDatabase, 115
  - createEmptyFile, 115
  - createResourcesDir, 116
  - enableHelpOnStartup, 116
  - event\_index\_, 136
  - exportProject, 116
  - exportTemplates, 117
  - FamilyTree, 111
  - file\_index\_, 136
  - getEvent, 117
  - getEvents, 117
  - getFile, 117
  - getFileAbsolutePath, 118
  - getFiles, 118
  - getGeneralOrphaFiles, 118
  - getMainPerson, 119
  - getOrphanFiles, 119
  - getParentSuggestions, 119
  - getPartnersSuggestions, 120
  - getPerson, 120
  - getPersons, 120
  - getProjectPaths, 120
  - getRelation, 120
  - getRelations, 121
  - getRelationSuggestions, 121
  - getSettings, 121
  - getSiblingsSuggestions, 121
  - getUrlFile, 122
  - importProject, 122
  - importRootDir, 122
  - importTemplates, 123
  - isDirectorySet, 123
  - isSaved, 123
  - loadDatabaseFile, 123
  - log, 124
  - mainPerson\_, 136
  - media\_index\_, 136
  - note\_index\_, 136
  - openDatabase, 124
  - openHelp, 136
  - openSingleFileDatabase, 125
  - parser\_, 136
  - person\_index\_, 136
  - printCss, 125
  - printEvents, 125
  - printFiles, 126
  - printHtml, 126
  - printHtmlPerson, 126
  - printMainPersonHtml, 126
  - printPeople, 127
  - printRelations, 127
  - projectPaths, 137
  - promoteRelation, 127
  - readJsonEvents, 127
  - readJsonFiles, 128
  - readJsonPeople, 128
  - readJsonRelations, 128
  - relation\_index\_, 137
  - removeBackup, 129
  - removeEvent, 129
  - removeEventTemplate, 129
  - removeGeneralOrphanFile, 129
  - removeOrphanFiles, 130
  - removePerson, 130

- removeRelation, 130
- removeRelationTemplate, 130
- renameFile, 130
- restoreBackup, 131
- savedChanges\_, 137
- setMainPerson, 131
- setRelation, 132
- settings\_, 137
- setUnsaved, 132
- showHelpOnStartup, 132
- storeDatabase, 132
- storeFileDatabase, 132
- templatesBasedOnEventTemplate, 134
- templatesBasedOnRelationTemplate, 134
- updateEventsWithTemplate, 134
- writeConfig, 135
- FATHER
  - jsonlabel, 42
  - relation, 57
- father\_
  - Person, 213
- FATHERHOOD
  - relation, 57
- Fatherhood
  - config.h, 285
- FATHERHOOD\_DESC
  - relation, 57
- fatherhoods
  - SuggestionsDialog, 261
- Female
  - person.h, 304
- File, 137
  - File, 138
  - filename\_, 141
  - getFilename, 138
  - getId, 138
  - getRealName, 139
  - id\_, 141
  - readJson, 139
  - setFilename, 139
  - setId, 139
- file\_index\_
  - FamilyTree, 136
- fileId\_
  - FileTreeItem, 146
- filename\_
  - File, 141
- FILES
  - jsonlabel, 42
- Files, Notes & Media, 7
- files\_
  - VirtualDrive, 269
- FILES\_DIR
  - parser, 51
- FileTreeItem, 141
  - addFile, 144
  - addSubDrive, 144
  - drive\_, 146
  - fileId\_, 146
  - FileTreeItem, 142
  - FT\_, 146
  - getDrive, 144
  - getId, 144
  - isDir\_, 146
  - isDirectory, 145
  - removeFile, 145
  - removeFolder, 145
  - rename, 145
  - type\_, 146
- FileType
  - person.h, 304
- filterPersons
  - ChoosePersonDialog, 73
- filterProjectItems
  - MainWindow, 170
- findPerson
  - MainWindow, 170
- FIRST\_NAME
  - jsonlabel, 42
- FIRST\_PERSON
  - jsonlabel, 42
- firstName\_
  - RelationTemplate, 244
- fitTreeInView
  - MainWindow, 170
- font\_
  - MainWindow, 185
  - PersonsGraphicsItem, 220
  - VisualSettings, 274
- FONT\_FAMILY
  - jsonlabel, 42
- FONT\_SIZE
  - jsonlabel, 42
- fontFamily
  - AppSettings, 66
- fontSize
  - AppSettings, 66
- FORBIDDEN\_CHARS
  - error, 27
- FORMER\_HUSBAND
  - events, 30
- FORMER\_WIFE
  - events, 30
- FRONT\_TITLE
  - jsonlabel, 43
- FT
  - MainWindow, 186
- FT\_
  - AddFileDialog, 63
  - ChoosePersonDialog, 74
  - DateDialog, 82
  - EventDialog, 93
  - EventTemplatesDialog, 98
  - ExportProjectDialog, 106
  - ExportTemplatesDialog, 108
  - FileTreeItem, 146



- HelpDialog, [149](#)
- HtmlExport, [152](#)
- RelationDialog, [234](#)
- RelationItem, [237](#)
- RelationTemplateDialog, [248](#)
- SuggestionsDialog, [261](#)
- TagChooserDialog, [264](#)
- FUNERAL
  - events, [30](#)
- GEN\_DIFF
  - jsonlabel, [43](#)
- GEN\_SIZE\_DOWN
  - jsonlabel, [43](#)
- GEN\_SIZE\_UP
  - jsonlabel, [43](#)
- GENDER
  - html, [35](#)
  - jsonlabel, [43](#)
- Gender
  - person.h, [304](#)
- gender, [31](#)
  - AllGenders, [32](#)
  - str, [32](#)
- gender\_
  - Person, [213](#)
- GENERAL\_FILE
  - person.h, [304](#)
- GENERAL\_INFO
  - html, [35](#)
- generalGenerationNumber
  - MainWindow, [170](#)
- generationDiff\_
  - RelationTemplate, [244](#)
- genSizeDown
  - AppSettings, [66](#)
- genSizeDown\_
  - MainWindow, [186](#)
- genSizeUp
  - AppSettings, [66](#)
- genSizeUp\_
  - MainWindow, [186](#)
- getAbsolutePath
  - Parser, [192](#)
- getAfterTitle
  - Person, [201](#)
- getAppSettings
  - Settings, [252](#)
- getBirthDate
  - Person, [201](#)
- getBirthPlace
  - Person, [201](#)
- getDate
  - Event, [84](#)
- getDay
  - Date, [78](#)
- getDeathDate
  - Person, [202](#)
- getDeathPlace
  - Person, [202](#)
- getDescription
  - EventTemplate, [100](#)
  - RelationTemplate, [241](#)
- getDrive
  - FileTreeItem, [144](#)
- getEvent
  - FamilyTree, [117](#)
- getEvents
  - FamilyTree, [117](#)
  - Person, [202](#)
- getEventTemplate
  - Settings, [253](#)
- getEventTemplates
  - Settings, [253](#)
- getFather
  - Person, [202](#)
- getFile
  - FamilyTree, [117](#)
- getFileAbsolutePath
  - FamilyTree, [118](#)
- getFilename
  - File, [138](#)
- getFiles
  - FamilyTree, [118](#)
  - VirtualDrive, [266](#)
- getFilesRoot
  - Person, [202](#)
- getFilesRootPointer
  - Person, [203](#)
- getFileUrl
  - Parser, [192](#)
- getFirstDate
  - WrappedDate, [276](#)
- getFirstName
  - RelationTemplate, [241](#)
- getFirstPerson
  - Relation, [226](#)
- getFrontTitle
  - Person, [203](#)
- getFullName
  - Person, [203](#)
- getGender
  - Person, [203](#)
- getGeneralOrphaFiles
  - FamilyTree, [118](#)
- getGenerationDifference
  - RelationTemplate, [241](#)
- getGlobalMainPerson
  - Settings, [253](#)
- getId
  - AddFileDialog, [63](#)
  - Event, [84](#)
  - EventRoleItem, [94](#)
  - EventTemplate, [100](#)
  - File, [138](#)
  - FileTreeItem, [144](#)
  - Person, [204](#)

- ProjectItem, [224](#)
- Relation, [226](#)
- RelationTemplate, [241](#)
- getMaidenName
  - Person, [204](#)
- getMainPerson
  - FamilyTree, [119](#)
- getMonth
  - Date, [78](#)
- getMother
  - Person, [204](#)
- getName
  - Person, [204](#)
  - VirtualDrive, [266](#)
- getOrphanFiles
  - FamilyTree, [119](#)
- getParentSuggestions
  - FamilyTree, [119](#)
- getPartner
  - Person, [204](#)
- getPartnersSuggestions
  - FamilyTree, [120](#)
- getPerson
  - FamilyTree, [120](#)
- getPersons
  - Event, [85](#)
  - FamilyTree, [120](#)
- getPlace
  - Event, [85](#)
- getProjectPaths
  - FamilyTree, [120](#)
- getRealName
  - File, [139](#)
- getRelation
  - FamilyTree, [120](#)
  - RelationItem, [236](#)
- getRelations
  - FamilyTree, [121](#)
  - Person, [205](#)
- getRelationSuggestions
  - FamilyTree, [121](#)
- getRelationTemplate
  - Settings, [253](#)
- getRelationTemplates
  - Settings, [254](#)
- getRoles
  - EventTemplate, [100](#)
- getSecondDate
  - WrappedDate, [277](#)
- getSecondName
  - RelationTemplate, [241](#)
- getSecondPerson
  - Relation, [226](#)
- getSettings
  - Event, [85](#)
  - FamilyTree, [121](#)
- getSiblingsSuggestions
  - FamilyTree, [121](#)
- getSubdrives
  - VirtualDrive, [267](#)
- getSurname
  - Person, [205](#)
- getTags
  - Person, [205](#)
  - Settings, [254](#)
- getTemplate
  - Event, [85](#)
  - Relation, [227](#)
- getText
  - Event, [85](#)
  - WrappedDate, [277](#)
- getTheOtherPerson
  - Relation, [227](#)
- getTheOtherPersonName
  - Relation, [227](#)
- getTitle
  - EventTemplate, [101](#)
  - Relation, [228](#)
  - RelationTemplate, [242](#)
- getTrait
  - Relation, [228](#)
  - RelationTemplate, [242](#)
- getUrlFile
  - FamilyTree, [122](#)
- getYear
  - Date, [78](#)
- GLOBAL\_MAIN\_PERSON
  - jsonlabel, [43](#)
- globalMainPerson\_
  - Settings, [258](#)
- GRADUATE
  - events, [30](#)
- GRADUATION
  - events, [31](#)
- graphics/dialogs.cpp, [318](#)
- graphics/dialogs.h, [318](#), [320](#)
- graphics/graphics\_items.cpp, [323](#)
- graphics/graphics\_items.h, [324](#), [326](#)
- graphics/mainwindow.cpp, [327](#)
- graphics/mainwindow.h, [328](#)
- graphics/mainwindow\_slots.cpp, [331](#)
- graphics/resources/icons.md, [331](#)
- graphics\_items.cpp
  - containsForb, [324](#)
- graphics\_items.h
  - containsForb, [325](#)
  - PROJECT\_EVENTS, [325](#)
  - PROJECT\_FILE, [325](#)
  - PROJECT\_GROUP, [325](#)
  - PROJECT\_MEDIA, [325](#)
  - PROJECT\_NOTE, [325](#)
  - PROJECT\_PERSON, [325](#)
  - PROJECT\_RELATIONS, [325](#)
  - ProjectItemType, [325](#)
- GraphicsView, [147](#)
- GraphicsView, [147](#)

- wheelEvent, [147](#)
- GREEN
  - jsonlabel, [43](#)
- green
  - Color, [76](#)
- GROOM
  - events, [31](#)
- hasMorePeopleInvolved
  - EventTemplate, [101](#)
- hasRoles
  - EventTemplate, [101](#)
- HELP
  - jsonlabel, [43](#)
- help texts/events.md, [331](#)
- help texts/files.md, [331](#)
- help texts/generic.md, [331](#)
- help texts/layout.md, [331](#)
- help texts/persons.md, [331](#)
- help texts/relations.md, [331](#)
- HelpDialog, [148](#)
  - ~HelpDialog, [149](#)
  - FT\_, [149](#)
  - HelpDialog, [148](#)
  - openHelp, [149](#)
  - ui, [149](#)
- highlighted\_
  - PersonsGraphicsItem, [220](#)
- HIGHLIGHTED\_COLOR
  - jsonlabel, [44](#)
- highlightedColor
  - AppSettings, [66](#)
  - VisualSettings, [272](#)
- highlightedColor\_
  - MainWindow, [186](#)
  - VisualSettings, [274](#)
- hovered\_
  - PersonsGraphicsItem, [220](#)
- hoverEnterEvent
  - PersonsGraphicsItem, [218](#)
- hoverLeaveEvent
  - PersonsGraphicsItem, [218](#)
- html, [32](#)
  - ALL\_PERSONS, [34](#)
  - BIRTH\_DATE, [34](#)
  - BIRTH\_PLACE, [34](#)
  - CLOSE\_FAMILY, [34](#)
  - CLOSE\_FAMILY\_DIAGRAM, [34](#)
  - CUSTOM\_CSS, [34](#)
  - CUSTOM\_TAGS, [34](#)
  - DATE, [34](#)
  - DEATH\_DATE, [35](#)
  - DEATH\_PLACE, [35](#)
  - DESCRIPTION, [35](#)
  - EVENTS, [35](#)
  - FAMILY\_TREE, [35](#)
  - FAMILY\_TREE\_DIAGRAM, [35](#)
  - GENDER, [35](#)
  - GENERAL\_INFO, [35](#)
  - NOTE, [36](#)
  - PERSONS, [36](#)
  - PLACE, [36](#)
  - PREFIX\_CLOSE, [36](#)
  - PREFIX\_FAMILY\_TREE, [36](#)
  - RELATIONS, [36](#)
  - RESOURCES, [36](#)
- HtmlExport, [150](#)
  - ~HtmlExport, [151](#)
  - allAboveGenerations, [151](#)
  - allBelowGenerations, [151](#)
  - exportAllHtml, [151](#)
  - exportMainHtml, [151](#)
  - FT\_, [152](#)
  - HtmlExport, [150](#)
  - ui, [152](#)
- HUSBAND
  - relation, [57](#)
- icons, [3](#)
- ID
  - jsonlabel, [44](#)
- id\_
  - Event, [88](#)
  - EventRoleItem, [95](#)
  - EventTemplate, [104](#)
  - File, [141](#)
  - Person, [213](#)
  - ProjectItem, [225](#)
  - Relation, [231](#)
  - RelationTemplate, [245](#)
- importJson
  - Person, [205](#)
- importProject
  - FamilyTree, [122](#)
  - MainWindow, [171](#)
- importRootDir
  - FamilyTree, [122](#)
  - MainWindow, [171](#)
- importTemplates
  - FamilyTree, [123](#)
  - MainWindow, [171](#)
  - Settings, [254](#)
- index.md, [331](#)
- indexes
  - Settings, [254](#)
- initializeProjectSubView
  - MainWindow, [171](#)
- initializeProjectView
  - MainWindow, [172](#)
- interractWithProjectItem
  - MainWindow, [172](#)
- isAlive
  - Person, [206](#)
- isDir\_
  - FileTreeItem, [146](#)
- isDirectory
  - FileTreeItem, [145](#)
- isDirectorySet

- FamilyTree, [123](#)
- isEmpty
  - Date, [78](#)
- isExact
  - Date, [78](#)
  - WrappedDate, [277](#)
- isOlderGeneration
  - Relation, [228](#)
- isParent
  - Relation, [228](#)
- isRootDirectorySet
  - Parser, [192](#)
- isSameGeneration
  - Relation, [229](#)
- isSaved
  - FamilyTree, [123](#)
- isUnknown
  - WrappedDate, [277](#)
- isYoungerGeneration
  - Relation, [229](#)
- itemClicked
  - PersonsGraphicsItem, [219](#)
- JSON\_CONFIG
  - parser, [51](#)
- JSON\_EVENTS
  - parser, [51](#)
- JSON\_FILES
  - parser, [51](#)
- JSON\_MEDIA
  - parser, [51](#)
- JSON\_NOTES
  - parser, [51](#)
- JSON\_PERSONS
  - parser, [52](#)
- JSON\_RELATIONS
  - parser, [52](#)
- jsonlabel, [37](#)
  - AFTER\_TITLE, [40](#)
  - APP\_SETTINGS, [40](#)
  - BIRTH, [40](#)
  - BLUE, [40](#)
  - BORDER\_RADIUS, [40](#)
  - CONTAINS\_DATE, [41](#)
  - CONTAINS\_PLACE, [41](#)
  - DATE, [41](#)
  - DEATH, [41](#)
  - DESCRIPTION, [41](#)
  - DIR, [41](#)
  - DRIVES, [41](#)
  - EVE\_TEMPL, [41](#)
  - EVENTS, [42](#)
  - EXPORT, [42](#)
  - FATHER, [42](#)
  - FILES, [42](#)
  - FIRST\_NAME, [42](#)
  - FIRST\_PERSON, [42](#)
  - FONT\_FAMILY, [42](#)
  - FONT\_SIZE, [42](#)
  - FRONT\_TITLE, [43](#)
  - GEN\_DIFF, [43](#)
  - GEN\_SIZE\_DOWN, [43](#)
  - GEN\_SIZE\_UP, [43](#)
  - GENDER, [43](#)
  - GLOBAL\_MAIN\_PERSON, [43](#)
  - GREEN, [43](#)
  - HELP, [43](#)
  - HIGHLIGHTED\_COLOR, [44](#)
  - ID, [44](#)
  - LAST\_DATE, [44](#)
  - LINE\_COLOR, [44](#)
  - LINE\_WIDTH, [44](#)
  - LIVES, [44](#)
  - MAIDEN\_NAME, [44](#)
  - MEDIA, [44](#)
  - MORE\_PEOPLE, [45](#)
  - MOTHER, [45](#)
  - MOVEABLE\_TABS, [45](#)
  - NAME, [45](#)
  - NOTES, [45](#)
  - PARTNER, [45](#)
  - PERSON, [45](#)
  - PERSONS, [45](#)
  - PLACE, [46](#)
  - PROBANT\_COLOR, [46](#)
  - PROMOTED\_COLOR, [46](#)
  - RED, [46](#)
  - REL\_TEMPL, [46](#)
  - RELATIONS, [46](#)
  - ROLE, [46](#)
  - ROOT, [46](#)
  - SECOND\_NAME, [47](#)
  - SECOND\_PERSON, [47](#)
  - SHOW\_EVENT\_TAB, [47](#)
  - SHOW\_FILE\_TAB, [47](#)
  - SHOW\_GENERAL\_TAB, [47](#)
  - SHOW\_MEDIA\_TAB, [47](#)
  - SHOW\_NOTE\_TAB, [47](#)
  - SHOW\_RELATION\_TAB, [47](#)
  - SPECIAL\_ROLES, [48](#)
  - SPLITTER\_POSITION\_ONE, [48](#)
  - SPLITTER\_POSITION\_TWO, [48](#)
  - STANDARD\_COLOR, [48](#)
  - SURNAME, [48](#)
  - TAB\_POSITION, [48](#)
  - TAG, [48](#)
  - TAGS, [48](#)
  - TEMPLATE, [49](#)
  - TEMPLATES, [49](#)
  - TEXT, [49](#)
  - TEXT\_COLOR, [49](#)
  - TITLE, [49](#)
  - TRAIT, [49](#)
  - TYPE, [49](#)
  - VALUE, [49](#)
  - VIRTUAL\_DRIVE, [50](#)
  - X\_SIZE, [50](#)

- Y\_SIZE, 50
- LAST\_DATE
  - jsonlabel, 44
- Layout, 11
- LINE\_COLOR
  - jsonlabel, 44
- LINE\_WIDTH
  - jsonlabel, 44
- lineColor
  - AppSettings, 66
  - VisualSettings, 272
- linePen\_
  - MainWindow, 186
- lineWidth
  - AppSettings, 66
- LIVES
  - jsonlabel, 44
- lives\_
  - Person, 213
- loadDatabaseFile
  - FamilyTree, 123
- loadSettings
  - MainWindow, 172
- lockedPerson
  - RelationSuggestion, 239
- log
  - FamilyTree, 124
  - Parser, 192
- MAID\_OF\_HONOR
  - events, 31
- MAIDEN\_NAME
  - jsonlabel, 44
- maidenName\_
  - Person, 214
- main
  - main.cpp, 332
- main.cpp, 331
  - main, 332
- main\_
  - EventDialog, 93
- mainPerson\_
  - FamilyTree, 136
- MainWindow, 152
  - ~MainWindow, 158
  - addGeneralFile, 159
  - addGeneralFolder, 159
  - addNewEvent, 159
  - addNewFile, 159
  - addNewFileFolder, 159
  - addNewMedia, 160
  - addNewMediaFolder, 160
  - addNewNote, 160
  - addNewNoteFolder, 160
  - addNewPerson, 160
  - addNewRelation, 160
  - allAboveGenerations, 160
  - allBelowGenerations, 161
  - borderRadius\_, 185
  - changeGenerationsDown, 161
  - changeGenerationsUp, 161
  - checkStoreRefreshPersonsInfo, 161
  - clearProjectFilter, 161
  - clearProjectPaths, 162
  - clearUi, 162
  - closeEvent, 162
  - connectAllSlots, 162
  - disableFileFolderPrompt, 162
  - disableMediaFolderPrompt, 162
  - disableNoteFolderPrompt, 162
  - drawCloseContainer, 163
  - drawCloseFamily, 163
  - drawFamilyTree, 164
  - drawGeneralGenerations, 164
  - drawOlderGenerations, 165
  - drawPartner, 165
  - drawYoungerGenerations, 166
  - editEvent, 166
  - editRelation, 167
  - editTag, 167
  - enableFileFolderPrompt, 167
  - enableGeneralFolderPrompt, 167
  - enableGeneralFolderRenamePrompt, 168
  - enableMediaFolderPrompt, 168
  - enableNoteFolderPrompt, 168
  - exportCloseFamilyDiagram, 168
  - exportCloseFamilyDiagramDialog, 169
  - exportGeneralDiagram, 169
  - exportHtml, 169
  - exportMainHtml, 169
  - exportProject, 169
  - exportTemplates, 169
  - exportTreeDiagram, 170
  - exportTreeDiagramDialog, 170
  - filterProjectItems, 170
  - findPerson, 170
  - fitTreeInView, 170
  - font\_, 185
  - FT, 186
  - generalGenerationNumber, 170
  - genSizeDown\_, 186
  - genSizeUp\_, 186
  - highlightedColor\_, 186
  - importProject, 171
  - importRootDir, 171
  - importTemplates, 171
  - initializeProjectSubView, 171
  - initializeProjectView, 172
  - interractWithProjectItem, 172
  - linePen\_, 186
  - loadSettings, 172
  - MainWindow, 158
  - olderGenerationNumber, 172
  - openBirthDateDialog, 173
  - openCustomDateDialog, 173
  - openDeathDateDialog, 173

- openEventTemplatesDialog, 173
- openExistingProject, 173
- openExternalFile, 173
- openGeneralFile, 174
- openGeneralProject, 174
- openGraphicSettings, 174
- openNewFamilyTreeProject, 174
- openRelationTemplatesDialog, 174
- openSelectedFile, 174
- openSelectedMedia, 175
- openSelectedNote, 175
- probandColor\_, 186
- promotedColor\_, 186
- promoteRelation, 175
- refreshDates, 175
- refreshEventTab, 175
- refreshGraphics, 176
- refreshPersonsInfoTab, 176
- refreshProjectEventsView, 176
- refreshProjectFilesView, 176
- refreshProjectGeneralFilesView, 176
- refreshProjectMediaView, 177
- refreshProjectNoteView, 177
- refreshProjectRelationsView, 177
- refreshProjectView, 178
- refreshRelationTab, 178
- refreshTabsWithFiles, 178
- refreshTagsList, 178
- refreshUi, 178
- removeEvent, 178
- removeFileFolder, 178
- removeGeneralFolder, 179
- removeMediaFolder, 179
- removeNoteFolder, 179
- removePerson, 179
- removeRelation, 179
- removeTag, 179
- renameFile, 179
- renameGeneralFileItem, 180
- renameMedia, 180
- renameNote, 180
- saveDatabase, 180
- saveGraphicsSetting, 180
- savePersonsInfo, 181
- scene, 186
- selectOlderGenerationOnly, 181
- selectSameGenerationOnly, 181
- selectYoungerGenerationOnly, 181
- setEnabledDeathAttributes, 181
- setEventTabVisible, 182
- setFileTabVisible, 182
- setGeneralTabVisible, 182
- setMainPerson, 182
- setMediaTabVisible, 182
- setNoteTabVisible, 182
- setNumberOfGenerations, 182
- setRelationTabVisible, 183
- setTabEast, 183
- setTabNorth, 183
- setTabSouth, 183
- setTabWest, 183
- setUiElements, 183
- showGeneralFiles, 184
- showHelp, 184
- showRenameFile, 184
- showRenameMedia, 184
- showRenameNote, 184
- sizeX\_, 187
- sizeY\_, 187
- splitterChanged, 184
- standardColor\_, 187
- switchNewTreeProject, 185
- textPen\_, 187
- treeScene, 187
- treeView, 187
- ui, 187
- youngerGenerationNumber, 185
- zoomIn, 185
- zoomOut, 185
- makeNewDatabase
  - Parser, 193
- Male
  - person.h, 304
- MARRIAGE
  - relation, 57
- MARRIAGE\_DESC
  - relation, 57
- MEDIA
  - jsonlabel, 44
  - person.h, 304
- MEDIA\_DIR
  - parser, 52
- media\_index\_
  - FamilyTree, 136
- month
  - Date, 80
- MORE\_PEOPLE
  - jsonlabel, 45
- morePeopleInvolved\_
  - EventTemplate, 104
- MOTHER
  - jsonlabel, 45
  - relation, 57
- mother\_
  - Person, 214
- MOTHERHOOD
  - relation, 58
- Motherhood
  - config.h, 285
- MOTHERHOOD\_DESC
  - relation, 58
- motherhoods
  - SuggestionsDialog, 261
- mousePressEvent
  - PersonsGraphicsItem, 219
- MOVEABLE\_TABS

- jsonlabel, 45
- NAME
  - jsonlabel, 45
- name\_
  - Person, 214
  - VirtualDrive, 269
- NAME\_STR
  - person, 53
- newOne\_
  - TagChooserDialog, 264
- NO\_CHOSEN\_FILE
  - error, 27
- NO\_ROLE
  - error, 28
  - events, 31
- NONE
  - addfile, 25
- None
  - config.h, 285
- North
  - config.h, 285
- NOTE
  - html, 36
  - person.h, 304
- note\_index\_
  - FamilyTree, 136
- NOTES
  - jsonlabel, 45
- NOTES\_DIR
  - parser, 52
- olderGenerationNumber
  - MainWindow, 172
- openBirthDateDialog
  - MainWindow, 173
- openCustomDate
  - EventDialog, 91
- openCustomDateDialog
  - MainWindow, 173
- openDatabase
  - FamilyTree, 124
- openDeathDateDialog
  - MainWindow, 173
- openEventTemplatesDialog
  - MainWindow, 173
- openExistingProject
  - MainWindow, 173
- openExternalFile
  - MainWindow, 173
- openFile
  - AddFileDialog, 63
- openGeneralFile
  - MainWindow, 174
- openGeneralProject
  - MainWindow, 174
- openGraphicSettings
  - MainWindow, 174
- openHelp
  - FamilyTree, 136
  - HelpDialog, 149
- openNewFamilyTreeProject
  - MainWindow, 174
- openRelationTemplatesDialog
  - MainWindow, 174
- openSelectedFile
  - MainWindow, 174
- openSelectedMedia
  - MainWindow, 175
- openSelectedNote
  - MainWindow, 175
- openSingleFileDatabase
  - FamilyTree, 125
- operator<<
  - config.cpp, 281–283
  - config.h, 285, 286
  - date.cpp, 289, 290
  - date.h, 291
  - family\_tree\_items.cpp, 295, 296
  - family\_tree\_items.h, 297
  - person.cpp, 302
  - person.h, 305
- Other
  - person.h, 304
- paint
  - CellGraphicsItem, 70
  - PersonsGraphicsItem, 219
- parent
  - RelationSuggestion, 239
- Parser, 188
  - checkFileConsistency, 189
  - clear, 189
  - configPath\_, 196
  - containsBackup, 190
  - containsBackupFile, 190
  - copyFile, 190
  - createAllDirs, 191
  - createEmptyFile, 191
  - createJsonFile, 191
  - createResourcesDir, 191
  - getAbsolutePath, 192
  - getBaseUrl, 192
  - isRootDirectorySet, 192
  - log, 192
  - makeNewDatabase, 193
  - Parser, 189
  - readConfig, 193
  - readJSONFile, 193
  - removeBackup, 194
  - removeFile, 194
  - renameFile, 194
  - restoreBackup, 194
  - root\_, 197
  - setDatabase, 195
  - writeConfig, 195
  - writeHtml, 195, 196
  - writeJSON, 196

- parser, 50
  - ERROR\_LOG, 51
  - FILES\_DIR, 51
  - JSON\_CONFIG, 51
  - JSON\_EVENTS, 51
  - JSON\_FILES, 51
  - JSON\_MEDIA, 51
  - JSON\_NOTES, 51
  - JSON\_PERSONS, 52
  - JSON\_RELATIONS, 52
  - MEDIA\_DIR, 52
  - NOTES\_DIR, 52
- parser\_
  - FamilyTree, 136
- PARTNER
  - jsonlabel, 45
  - relation, 58
- partner\_
  - Person, 214
- PARTNERSHIP
  - relation, 58
- Partnership
  - config.h, 285
- PARTNERSHIP\_DESC
  - relation, 58
- partnerships
  - SuggestionsDialog, 261
- pen\_
  - CellGraphicsItem, 71
  - PersonsGraphicsItem, 220
  - VisualSettings, 274
- PERSON
  - jsonlabel, 45
- Person, 13, 197
  - addEvent, 199
  - addRelation, 199
  - addTag, 200
  - birthDate, 200
  - dateOfBirth\_, 213
  - dateOfDeath\_, 213
  - deathDate, 200
  - events\_, 213
  - existsTag, 200
  - exportJson, 201
  - father\_, 213
  - gender\_, 213
  - getAfterTitle, 201
  - getBirthDate, 201
  - getBirthPlace, 201
  - getDeathDate, 202
  - getDeathPlace, 202
  - getEvents, 202
  - getFather, 202
  - getFilesRoot, 202
  - getFilesRootPointer, 203
  - getFrontTitle, 203
  - getFullName, 203
  - getGender, 203
  - getId, 204
  - getMaidenName, 204
  - getMother, 204
  - getName, 204
  - getPartner, 204
  - getRelations, 205
  - getSurname, 205
  - getTags, 205
  - id\_, 213
  - importJson, 205
  - isAlive, 206
  - lives\_, 213
  - maidenName\_, 214
  - mother\_, 214
  - name\_, 214
  - partner\_, 214
  - Person, 199
  - placeOfBirth\_, 214
  - placeOfDeath\_, 214
  - printHtml, 206
  - promoteRelation, 206
  - readJson, 206
  - relations\_, 214
  - removeEvent, 207
  - removeRelation, 207
  - removeTag, 207
  - rootFileDrive\_, 214
  - rootMediaDrive\_, 215
  - rootNoteDrive\_, 215
  - setAfterTitle, 207
  - setBirthDate, 208
  - setBirthPlace, 208
  - setDeathDate, 208
  - setDeathPlace, 208
  - setFather, 209
  - setFilesRoot, 209
  - setFrontTitle, 209
  - setGender, 209
  - setId, 210
  - setLives, 210
  - setMaidenName, 210
  - setMother, 210
  - setName, 211
  - setPartner, 211
  - setSurname, 211
  - str, 211
  - surname\_, 215
  - tags\_, 215
  - titleAfter\_, 215
  - titleInFront\_, 215
  - updateSpecialRelation, 211
- person, 52
  - DATE\_FROM, 53
  - DATE\_TO, 53
  - NAME\_STR, 53
  - ROOT, 53
  - SURNAME\_STR, 53
  - UNKNOWN\_STR, 53



- person.cpp
  - operator<<, 302
  - printSizeTContainer, 303
- person.h
  - Female, 304
  - FileType, 304
  - Gender, 304
  - GENERAL\_FILE, 304
  - Male, 304
  - MEDIA, 304
  - NOTE, 304
  - operator<<, 305
  - Other, 304
  - printSizeTContainer, 305
- person1\_
  - Relation, 231
  - RelationDialog, 234
- person2\_
  - Relation, 231
  - RelationDialog, 234
- person\_
  - ChoosePersonDialog, 74
  - PersonsGraphicsItem, 220
- person\_index\_
  - FamilyTree, 136
- PERSONS
  - html, 36
  - jsonlabel, 45
- persons\_
  - Event, 88
- PersonsGraphicsItem, 216
  - ~PersonsGraphicsItem, 218
  - bold\_text\_, 220
  - borderRadius\_, 220
  - boundingRect, 218
  - color\_, 220
  - font\_, 220
  - highlighted\_, 220
  - hovered\_, 220
  - hoverEnterEvent, 218
  - hoverLeaveEvent, 218
  - itemClicked, 219
  - mousePressEvent, 219
  - paint, 219
  - pen\_, 220
  - person\_, 220
  - PersonsGraphicsItem, 217
  - shape, 219
  - size\_x\_, 221
  - size\_y\_, 221
  - textLines\_, 221
  - textPen\_, 221
  - x\_, 221
  - y\_, 221
- PictureExport, 222
  - ~PictureExport, 223
  - closeDiagram\_, 223
  - exportDiagram, 223
  - PictureExport, 222
  - ui, 223
- PLACE
  - html, 36
  - jsonlabel, 46
- place\_
  - Event, 89
- placeOfBirth\_
  - Person, 214
- placeOfDeath\_
  - Person, 214
- PREFIX\_CLOSE
  - html, 36
- PREFIX\_FAMILY\_TREE
  - html, 36
- printCss
  - FamilyTree, 125
- printEvents
  - FamilyTree, 125
- printFiles
  - FamilyTree, 126
- printHtml
  - Event, 86
  - FamilyTree, 126
  - Person, 206
  - Relation, 229
- printHtmlPerson
  - FamilyTree, 126
- printMainPersonHtml
  - FamilyTree, 126
- printPeople
  - FamilyTree, 127
- printRelations
  - FamilyTree, 127
- printSizeTContainer
  - person.cpp, 303
  - person.h, 305
- probandColor
  - AppSettings, 67
  - VisualSettings, 272
- probandColor\_
  - MainWindow, 186
  - VisualSettings, 274
- PROBANT\_COLOR
  - jsonlabel, 46
- PROJECT\_EVENTS
  - graphics\_items.h, 325
- PROJECT\_FILE
  - graphics\_items.h, 325
- PROJECT\_GROUP
  - graphics\_items.h, 325
- PROJECT\_MEDIA
  - graphics\_items.h, 325
- PROJECT\_NOTE
  - graphics\_items.h, 325
- PROJECT\_PERSON
  - graphics\_items.h, 325
- PROJECT\_RELATIONS

- graphics\_items.h, 325
- ProjectItem, 223
  - getId, 224
  - id\_, 225
  - ProjectItem, 224
  - type, 224
  - type\_, 225
- ProjectItemType
  - graphics\_items.h, 325
- projectPaths
  - FamilyTree, 137
- promoted
  - AppSettings, 67
- PROMOTED\_COLOR
  - jsonlabel, 46
- promotedColor
  - VisualSettings, 272
- promotedColor\_
  - MainWindow, 186
  - VisualSettings, 274
- promoteRelation
  - FamilyTree, 127
  - MainWindow, 175
  - Person, 206
- radius
  - AppSettings, 67
- read
  - Date, 78
- readConfig
  - Parser, 193
- readJson
  - AppSettings, 65
  - Color, 76
  - Event, 86
  - EventTemplate, 101
  - File, 139
  - Person, 206
  - Relation, 230
  - RelationTemplate, 242
  - Settings, 255
  - VirtualDrive, 267
  - WrappedDate, 277
- readJsonEvents
  - FamilyTree, 127
- readJSONFile
  - Parser, 193
- readJsonFiles
  - FamilyTree, 128
- readJsonPeople
  - FamilyTree, 128
- readJsonRelations
  - FamilyTree, 128
- RED
  - jsonlabel, 46
- red
  - Color, 76
- refresh
  - VisualSettings, 272
- refreshDates
  - EventDialog, 91
  - MainWindow, 175
- refreshEventTab
  - MainWindow, 175
- refreshGraphics
  - MainWindow, 176
- refreshListWidget
  - EventTemplatesDialog, 97
  - RelationTemplateDialog, 247
- refreshPersons
  - EventDialog, 91
- refreshPersonsInfoTab
  - MainWindow, 176
- refreshProjectEventsView
  - MainWindow, 176
- refreshProjectFilesView
  - MainWindow, 176
- refreshProjectGeneralFilesView
  - MainWindow, 176
- refreshProjectMediaView
  - MainWindow, 177
- refreshProjectNoteView
  - MainWindow, 177
- refreshProjectRelationsView
  - MainWindow, 177
- refreshProjectView
  - MainWindow, 178
- refreshRelationTab
  - MainWindow, 178
- refreshTabsWithFiles
  - MainWindow, 178
- refreshTagsList
  - MainWindow, 178
- refreshTexts
  - RelationItem, 236
- refreshUi
  - MainWindow, 178
- relIndex
  - Settings, 255
- rel\_
  - RelationDialog, 235
- REL\_TEMPL
  - jsonlabel, 46
- Relation, 225
  - getFirstPerson, 226
  - getId, 226
  - getSecondPerson, 226
  - getTemplate, 227
  - getTheOtherPerson, 227
  - getTheOtherPersonName, 227
  - getTitle, 228
  - getTrait, 228
  - id\_, 231
  - isOlderGeneration, 228
  - isParent, 228
  - isSameGeneration, 229
  - isYoungerGeneration, 229

- person1\_, 231
- person2\_, 231
- printHtml, 229
- readJson, 230
- Relation, 226
- setId, 230
- setPersons, 230
- setTemplate, 230
- settings\_, 231
- str, 230
- template\_, 231
- relation, 53
  - ADOPTED\_CHILD, 55
  - ADOPTIVE\_FATHER, 55
  - ADOPTIVE\_FATHERHOOD, 55
  - ADOPTIVE\_MOTHER, 55
  - ADOPTIVE\_MOTHERHOOD, 55
  - BROTHER, 55
  - BROTHERHOOD, 56
  - CHILD, 56
  - DESC\_ADOPTIVE\_FATHERHOOD, 56
  - DESC\_ADOPTIVE\_MOTHERHOOD, 56
  - DESC\_BROTHERHOOD, 56
  - DESC\_SIBLINGS, 56
  - DESC\_SISTERHOOD, 56
  - FATHER, 57
  - FATHERHOOD, 57
  - FATHERHOOD\_DESC, 57
  - HUSBAND, 57
  - MARRIAGE, 57
  - MARRIAGE\_DESC, 57
  - MOTHER, 57
  - MOTHERHOOD, 58
  - MOTHERHOOD\_DESC, 58
  - PARTNER, 58
  - PARTNERSHIP, 58
  - PARTNERSHIP\_DESC, 58
  - SIBLINGS, 58
  - SISTER, 58
  - SISTERHOOD, 58
  - WIFE, 59
- relation\_
  - RelationItem, 237
- relation\_index\_
  - FamilyTree, 137
- RelationDialog, 232
  - ~RelationDialog, 233
  - chooseFirstPerson, 233
  - chooseSecondPerson, 233
  - FT\_, 234
  - person1\_, 234
  - person2\_, 234
  - rel\_, 235
  - RelationDialog, 233
  - saveFirstPerson\_, 235
  - savePerson, 234
  - saveRelation, 234
  - showTemplateDetails, 234
  - switchPersons, 234
  - ui, 235
- RelationItem, 235
  - FT\_, 237
  - getRelation, 236
  - refreshTexts, 236
  - relation\_, 237
  - RelationItem, 236
- RELATIONS
  - html, 36
  - jsonlabel, 46
- Relations, 15
- relations\_
  - Person, 214
- relationsTemplates\_
  - Settings, 258
- RelationSuggestion, 237
  - lockedPerson, 239
  - parent, 239
  - RelationSuggestion, 238
  - suggestedPerson, 239
  - symmetrical, 239
  - trait, 239
- RelationTemplate, 240
  - description\_, 244
  - firstName\_, 244
  - generationDiff\_, 244
  - getDescription, 241
  - getFirstName, 241
  - getGenerationDifference, 241
  - getId, 241
  - getSecondName, 241
  - getTitle, 242
  - getTrait, 242
  - id\_, 245
  - readJson, 242
  - RelationTemplate, 241
  - secondName\_, 245
  - setDescription, 242
  - setFirstName, 243
  - setGenerationDifference, 243
  - setId, 243
  - setSecondName, 243
  - setTitle, 243
  - setTrait, 244
  - str, 244
  - title\_, 245
  - trait\_, 245
- RelationTemplateDialog, 245
  - ~RelationTemplateDialog, 247
  - addDefaults, 247
  - addingNew\_, 248
  - addNewTemplate, 247
  - clear, 247
  - FT\_, 248
  - refreshListWidget, 247
  - RelationTemplateDialog, 246
  - removeTemplate, 247

- saveTemplate, [247](#)
  - setEnabledWidgets, [248](#)
  - showTemplate, [248](#)
  - ui, [248](#)
- relationTemplateIndex\_
  - Settings, [258](#)
- REMOVE
  - button, [26](#)
- remove
  - VirtualDrive, [267](#)
- removeBackup
  - FamilyTree, [129](#)
  - Parser, [194](#)
- removeEmptyRole
  - EventDialog, [92](#)
- removeEvent
  - FamilyTree, [129](#)
  - MainWindow, [178](#)
  - Person, [207](#)
- removeEventTemplate
  - FamilyTree, [129](#)
  - Settings, [255](#)
- removeFile
  - FileTreeItem, [145](#)
  - Parser, [194](#)
  - VirtualDrive, [267](#)
- removeFileFolder
  - MainWindow, [178](#)
- removeFolder
  - FileTreeItem, [145](#)
- removeGeneralFolder
  - MainWindow, [179](#)
- removeGeneralOrphanFile
  - FamilyTree, [129](#)
- removeMediaFolder
  - MainWindow, [179](#)
- removeNoteFolder
  - MainWindow, [179](#)
- removeOrphanFiles
  - FamilyTree, [130](#)
- removePerson
  - Event, [86](#)
  - EventDialog, [92](#)
  - FamilyTree, [130](#)
  - MainWindow, [179](#)
- removeRelation
  - FamilyTree, [130](#)
  - MainWindow, [179](#)
  - Person, [207](#)
- removeRelationTemplate
  - FamilyTree, [130](#)
  - Settings, [255](#)
- removeRole
  - EventTemplatesDialog, [97](#)
- removeSuggestion
  - SuggestionsDialog, [260](#)
- removeTag
  - MainWindow, [179](#)
- Person, [207](#)
  - Settings, [257](#)
- removeTemplate
  - EventTemplatesDialog, [97](#)
  - RelationTemplateDialog, [247](#)
- rename
  - FileTreeItem, [145](#)
  - VirtualDrive, [269](#)
- renameFile
  - FamilyTree, [130](#)
  - MainWindow, [179](#)
  - Parser, [194](#)
- renameGeneralFileItem
  - MainWindow, [180](#)
- renameMedia
  - MainWindow, [180](#)
- renameNote
  - MainWindow, [180](#)
- RESOURCES
  - html, [36](#)
- restoreBackup
  - FamilyTree, [131](#)
  - Parser, [194](#)
- Rodoc, [1](#)
- ROLE
  - jsonlabel, [46](#)
- roles\_
  - EventTemplate, [104](#)
- ROOT
  - jsonlabel, [46](#)
  - person, [53](#)
- root\_
  - Parser, [197](#)
- rootFileDrive\_
  - Person, [214](#)
- rootMediaDrive\_
  - Person, [215](#)
- rootNoteDrive\_
  - Person, [215](#)
- save
  - SuggestionsDialog, [260](#)
- saveAndExit
  - VisualSettings, [273](#)
- saveDatabase
  - MainWindow, [180](#)
- saveDate
  - DateDialog, [81](#)
- savedChanges\_
  - FamilyTree, [137](#)
- saveEvent
  - EventDialog, [92](#)
- saveFirstPerson\_
  - RelationDialog, [235](#)
- saveGraphicsSetting
  - MainWindow, [180](#)
- savePerson
  - ChoosePersonDialog, [74](#)
  - EventDialog, [92](#)

- RelationDialog, [234](#)
- savePersonsInfo
  - MainWindow, [181](#)
- saveRelation
  - RelationDialog, [234](#)
- saveSelectedPerson
  - ChoosePersonDialog, [74](#)
- saveSettings
  - VisualSettings, [273](#)
- saveTag
  - TagChooserDialog, [263](#)
- saveTemplate
  - EventTemplatesDialog, [97](#)
  - RelationTemplateDialog, [247](#)
- scene
  - MainWindow, [186](#)
  - VisualSettings, [274](#)
- SECOND\_NAME
  - jsonlabel, [47](#)
- SECOND\_PERSON
  - jsonlabel, [47](#)
- secondName\_
  - RelationTemplate, [245](#)
- selectedSuggestion
  - SuggestionsDialog, [260](#)
- selectOlderGenerationOnly
  - MainWindow, [181](#)
- selectSameGenerationOnly
  - MainWindow, [181](#)
- selectYoungerGenerationOnly
  - MainWindow, [181](#)
- setAfterTitle
  - Person, [207](#)
- setAppSettings
  - Settings, [257](#)
- setBirthDate
  - Person, [208](#)
- setBirthPlace
  - Person, [208](#)
- setContainsDate
  - EventTemplate, [102](#)
- setContainsPlace
  - EventTemplate, [102](#)
- setDatabase
  - Parser, [195](#)
- setDate
  - Event, [87](#)
- setDay
  - Date, [79](#)
- setDeathDate
  - Person, [208](#)
- setDeathPlace
  - Person, [208](#)
- setDefault
  - AppSettings, [65](#)
- setDescription
  - EventTemplate, [102](#)
  - RelationTemplate, [242](#)
- setEnabledDeathAttributes
  - MainWindow, [181](#)
- setEnabledWidgets
  - EventTemplatesDialog, [98](#)
  - RelationTemplateDialog, [248](#)
- setEventTabVisible
  - MainWindow, [182](#)
- setFather
  - Person, [209](#)
- setFilename
  - File, [139](#)
- setFilesRoot
  - Person, [209](#)
- setFileTabVisible
  - MainWindow, [182](#)
- setFirstDate
  - WrappedDate, [278](#)
- setFirstName
  - RelationTemplate, [243](#)
- setFrontTitle
  - Person, [209](#)
- setGender
  - Person, [209](#)
- setGeneralTabVisible
  - MainWindow, [182](#)
- setGenerationDifference
  - RelationTemplate, [243](#)
- setGlobalMainPerson
  - Settings, [257](#)
- setId
  - Event, [87](#)
  - EventTemplate, [102](#)
  - File, [139](#)
  - Person, [210](#)
  - Relation, [230](#)
  - RelationTemplate, [243](#)
- setLives
  - Person, [210](#)
- setMaidenName
  - Person, [210](#)
- setMainPerson
  - FamilyTree, [131](#)
  - MainWindow, [182](#)
- setMediaTabVisible
  - MainWindow, [182](#)
- setMonth
  - Date, [79](#)
- setMorePeopleInvolved
  - EventTemplate, [103](#)
- setMother
  - Person, [210](#)
- setName
  - Person, [211](#)
- setNoteTabVisible
  - MainWindow, [182](#)
- setNumberOfGenerations
  - MainWindow, [182](#)
- setPartner

- Person, 211
- setPerson
  - EventRoleItem, 94
- setPersons
  - Relation, 230
- setPlace
  - Event, 87
- setRelation
  - FamilyTree, 132
- setRelationTabVisible
  - MainWindow, 183
- setRoles
  - EventTemplate, 103
- setSecondDate
  - WrappedDate, 278
- setSecondName
  - RelationTemplate, 243
- setSurname
  - Person, 211
- setTabEast
  - MainWindow, 183
- setTabNorth
  - MainWindow, 183
- setTabSouth
  - MainWindow, 183
- setTabWest
  - MainWindow, 183
- setTemplate
  - Event, 87
  - Relation, 230
- setText
  - Event, 88
  - WrappedDate, 278
- Settings, 249
  - addDefaultEvent, 250
  - addDefaultEvents, 251
  - addDefaultRelation, 251
  - addDefaultRelations, 251
  - addEventTemplate, 251
  - addRelationTemplate, 251
  - addTag, 252
  - appSettings\_, 257
  - clear, 252
  - eventTemplateIndex\_, 257
  - eventTemplates\_, 258
  - exportTemplates, 252
  - getAppSettings, 252
  - getEventTemplate, 253
  - getEventTemplates, 253
  - getGlobalMainPerson, 253
  - getRelationTemplate, 253
  - getRelationTemplates, 254
  - getTags, 254
  - globalMainPerson\_, 258
  - importTemplates, 254
  - indexes, 254
  - readJson, 255
  - reIndex, 255
  - relationsTemplates\_, 258
  - relationTemplateIndex\_, 258
  - removeEventTemplate, 255
  - removeRelationTemplate, 255
  - removeTag, 257
  - setAppSettings, 257
  - setGlobalMainPerson, 257
  - Settings, 250
  - tags\_, 258
- settings\_
  - Event, 89
  - FamilyTree, 137
  - Relation, 231
- setTitle
  - EventTemplate, 103
  - RelationTemplate, 243
- setTrait
  - RelationTemplate, 244
- setUiElements
  - MainWindow, 183
- setUnsaved
  - FamilyTree, 132
- setYear
  - Date, 79
- shape
  - CellGraphicsItem, 71
  - PersonsGraphicsItem, 219
- SHOW\_EVENT\_TAB
  - jsonlabel, 47
- SHOW\_FILE\_TAB
  - jsonlabel, 47
- SHOW\_GENERAL\_TAB
  - jsonlabel, 47
- SHOW\_MEDIA\_TAB
  - jsonlabel, 47
- SHOW\_NOTE\_TAB
  - jsonlabel, 47
- SHOW\_RELATION\_TAB
  - jsonlabel, 47
- showEventDetails
  - EventDialog, 92
- showEventTab
  - AppSettings, 67
- showFileTab
  - AppSettings, 67
- showGeneralFiles
  - MainWindow, 184
- showGeneralTab
  - AppSettings, 67
- showHelp
  - MainWindow, 184
- showHelpOnStartup
  - FamilyTree, 132
- showMediaTab
  - AppSettings, 67
- showNoteTab
  - AppSettings, 67
- showRelationTab

- AppSettings, 68
- showRenameFile
  - MainWindow, 184
- showRenameMedia
  - MainWindow, 184
- showRenameNote
  - MainWindow, 184
- showTemplate
  - EventDialog, 92
  - EventTemplatesDialog, 98
  - RelationTemplateDialog, 248
- showTemplateDetails
  - RelationDialog, 234
- Sibling
  - config.h, 285
- SIBLINGS
  - relation, 58
- siblings
  - SuggestionsDialog, 261
- SISTER
  - relation, 58
- SISTERHOOD
  - relation, 58
- sizeX
  - AppSettings, 68
- sizeX\_
  - MainWindow, 187
  - VisualSettings, 275
- sizeX\_
  - CellGraphicsItem, 71
  - PersonsGraphicsItem, 221
- sizeY
  - AppSettings, 68
- sizeY\_
  - MainWindow, 187
  - VisualSettings, 275
- sizeY\_
  - CellGraphicsItem, 71
  - PersonsGraphicsItem, 221
- South
  - config.h, 285
- SPECIAL\_ROLES
  - jsonlabel, 48
- SPLITTER\_POSITION\_ONE
  - jsonlabel, 48
- SPLITTER\_POSITION\_TWO
  - jsonlabel, 48
- splitterChanged
  - MainWindow, 184
- splitterPositionOne
  - AppSettings, 68
- splitterPositionTwo
  - AppSettings, 68
- standard
  - AppSettings, 68
- STANDARD\_COLOR
  - jsonlabel, 48
- standardColor
  - VisualSettings, 273
- standardColor\_
  - MainWindow, 187
  - VisualSettings, 275
- storeDatabase
  - FamilyTree, 132
- storeFileDatabase
  - FamilyTree, 132
- str
  - Date, 79
  - Event, 88
  - EventTemplate, 103
  - gender, 32
  - Person, 211
  - Relation, 230
  - RelationTemplate, 244
  - trait, 60
  - WrappedDate, 278
- strings.h
  - EMPTY\_STRING, 315
- subdrives\_
  - VirtualDrive, 269
- suggestedPerson
  - RelationSuggestion, 239
- SuggestionsDialog, 259
  - ~SuggestionsDialog, 260
  - fatherhoods, 261
  - FT\_, 261
  - motherhoods, 261
  - partnerships, 261
  - removeSuggestion, 260
  - save, 260
  - selectedSuggestion, 260
  - siblings, 261
  - SuggestionsDialog, 260
  - swapPersons, 261
  - switchedTemplate, 261
  - ui, 262
- SURNAME
  - jsonlabel, 48
- surname\_
  - Person, 215
- SURNAME\_STR
  - person, 53
- swapPersons
  - SuggestionsDialog, 261
- switchedTemplate
  - SuggestionsDialog, 261
- switchNewTreeProject
  - MainWindow, 185
- switchPersons
  - RelationDialog, 234
- symmetrical
  - RelationSuggestion, 239
- TAB\_POSITION
  - jsonlabel, 48
- TabPosition
  - config.h, 284

- tabPosition
  - AppSettings, 68
- TAG
  - jsonlabel, 48
- tagchooser, 59
  - CUSTOM\_TAG, 59
- TagChooserDialog, 262
  - ~TagChooserDialog, 263
  - changeVisibleWidgets, 263
  - FT\_, 264
  - newOne\_, 264
  - saveTag, 263
  - TagChooserDialog, 263
  - ui, 264
- TAGS
  - jsonlabel, 48
- tags\_
  - Person, 215
  - Settings, 258
- TEMPLATE
  - jsonlabel, 49
- template\_
  - Event, 89
  - Relation, 231
- TEMPLATES
  - jsonlabel, 49
- templatesBasedOnEventTemplate
  - FamilyTree, 134
- templatesBasedOnRelationTemplate
  - FamilyTree, 134
- TEXT
  - jsonlabel, 49
- text\_
  - Event, 89
  - WrappedDate, 279
- TEXT\_COLOR
  - jsonlabel, 49
- textColor
  - AppSettings, 68
  - VisualSettings, 273
- textLines\_
  - PersonsGraphicsItem, 221
- textPen\_
  - MainWindow, 187
  - PersonsGraphicsItem, 221
  - VisualSettings, 275
- TITLE
  - jsonlabel, 49
- title\_
  - EventTemplate, 104
  - RelationTemplate, 245
- titleAfter\_
  - Person, 215
- titleInFront\_
  - Person, 215
- TRAIT
  - jsonlabel, 49
- Trait
  - config.h, 285
- trait, 59
  - AllTraits, 60
  - RelationSuggestion, 239
  - str, 60
- trait\_
  - RelationTemplate, 245
- treeScene
  - MainWindow, 187
- treeView
  - MainWindow, 187
- TYPE
  - jsonlabel, 49
- type
  - ProjectItem, 224
- type\_
  - AddFileDialog, 63
  - FileTreeItem, 146
  - ProjectItem, 225
- Ui, 60
- ui
  - AddFileDialog, 64
  - ChoosePersonDialog, 74
  - DateDialog, 82
  - EventDialog, 93
  - EventTemplatesDialog, 98
  - ExportProjectDialog, 106
  - ExportTemplatesDialog, 108
  - HelpDialog, 149
  - HtmlExport, 152
  - MainWindow, 187
  - PictureExport, 223
  - RelationDialog, 235
  - RelationTemplateDialog, 248
  - SuggestionsDialog, 262
  - TagChooserDialog, 264
  - VisualSettings, 275
- UNKNOWN\_EXCEP\_STR
  - error, 28
- UNKNOWN\_STR
  - person, 53
- unknownDate
  - DateDialog, 81
- updateDate
  - WrappedDate, 279
- updateEventsWithTemplate
  - FamilyTree, 134
- updateSpecialRelation
  - Person, 211
- updateToTemplate
  - Event, 88
- usedFiles
  - VirtualDrive, 269
- VALUE
  - jsonlabel, 49
- VIRTUAL\_DRIVE
  - jsonlabel, 50



- VirtualDrive, 264
  - addFile, 265
  - addSubdrive, 265, 266
  - copyDrive, 266
  - files\_, 269
  - getFiles, 266
  - getName, 266
  - getSubdrives, 267
  - name\_, 269
  - readJson, 267
  - remove, 267
  - removeFile, 267
  - rename, 269
  - subdrives\_, 269
  - usedFiles, 269
  - VirtualDrive, 265
- VisualSettings, 270
  - ~VisualSettings, 272
  - borderRadius\_, 274
  - font\_, 274
  - highlightedColor, 272
  - highlightedColor\_, 274
  - lineColor, 272
  - pen\_, 274
  - probandColor, 272
  - probandColor\_, 274
  - promotedColor, 272
  - promotedColor\_, 274
  - refresh, 272
  - saveAndExit, 273
  - saveSettings, 273
  - scene, 274
  - sizeX\_, 275
  - sizeY\_, 275
  - standardColor, 273
  - standardColor\_, 275
  - textColor, 273
  - textPen\_, 275
  - ui, 275
  - VisualSettings, 271
- WEDDING
  - events, 31
- Welcome to rododoc, 9
- West
  - config.h, 285
- wheelEvent
  - GraphicsView, 147
- WIFE
  - relation, 59
- WrappedDate, 275
  - clear, 276
  - date1\_, 279
  - date2\_, 279
  - getFirstDate, 276
  - getSecondDate, 277
  - getText, 277
  - isExact, 277
  - isUnknown, 277
  - readJson, 277
  - setFirstDate, 278
  - setSecondDate, 278
  - setText, 278
  - str, 278
  - text\_, 279
  - updateDate, 279
  - WrappedDate, 276
- writeConfig
  - FamilyTree, 135
  - Parser, 195
- writeHtml
  - Parser, 195, 196
- writeJSON
  - Parser, 196
- x\_
  - CellGraphicsItem, 71
  - PersonsGraphicsItem, 221
- X\_SIZE
  - jsonlabel, 50
- y\_
  - CellGraphicsItem, 71
  - PersonsGraphicsItem, 221
- Y\_SIZE
  - jsonlabel, 50
- year
  - Date, 80
- youngerGenerationNumber
  - MainWindow, 185
- zoomIn
  - MainWindow, 185
- zoomOut
  - MainWindow, 185