

Csuper - Compteur de Score Universel Permettant l'Exemption de Reflexion

4.2.0

Generated by Doxygen 1.8.9.1

Fri Jul 24 2015 12:11:53

Contents

| | | |
|----------|---|----------|
| 1 | Data Structure Index | 1 |
| 1.1 | Data Structures | 1 |
| 2 | File Index | 3 |
| 2.1 | File List | 3 |
| 3 | Data Structure Documentation | 5 |
| 3.1 | calculatorList Struct Reference | 5 |
| 3.1.1 | Detailed Description | 5 |
| 3.1.2 | Field Documentation | 5 |
| 3.1.2.1 | current | 5 |
| 3.1.2.2 | first | 5 |
| 3.1.2.3 | last | 5 |
| 3.2 | calculatorNodeList Struct Reference | 5 |
| 3.2.1 | Detailed Description | 6 |
| 3.2.2 | Field Documentation | 6 |
| 3.2.2.1 | is_number | 6 |
| 3.2.2.2 | next | 6 |
| 3.2.2.3 | number | 6 |
| 3.2.2.4 | operator | 6 |
| 3.2.2.5 | previous | 6 |
| 3.3 | chart_exportation Struct Reference | 6 |
| 3.3.1 | Detailed Description | 7 |
| 3.3.2 | Field Documentation | 7 |
| 3.3.2.1 | height | 7 |
| 3.3.2.2 | total_points | 7 |
| 3.3.2.3 | width | 7 |
| 3.4 | csuStruct Struct Reference | 7 |
| 3.4.1 | Detailed Description | 7 |
| 3.4.2 | Field Documentation | 7 |
| 3.4.2.1 | config | 7 |
| 3.4.2.2 | day | 8 |

| | | |
|----------|--|----|
| 3.4.2.3 | distributor | 8 |
| 3.4.2.4 | month | 8 |
| 3.4.2.5 | nb_player | 8 |
| 3.4.2.6 | nb_turn | 8 |
| 3.4.2.7 | player_names | 8 |
| 3.4.2.8 | point | 8 |
| 3.4.2.9 | rank | 8 |
| 3.4.2.10 | size_max_name | 8 |
| 3.4.2.11 | total_points | 8 |
| 3.4.2.12 | version | 8 |
| 3.4.2.13 | year | 8 |
| 3.5 | difference_between_player Struct Reference | 9 |
| 3.5.1 | Detailed Description | 9 |
| 3.5.2 | Field Documentation | 9 |
| 3.5.2.1 | consecutive | 9 |
| 3.5.2.2 | first | 9 |
| 3.5.2.3 | last | 9 |
| 3.6 | export_pdf Struct Reference | 9 |
| 3.6.1 | Detailed Description | 9 |
| 3.6.2 | Field Documentation | 10 |
| 3.6.2.1 | font | 10 |
| 3.6.2.2 | line | 10 |
| 3.6.2.3 | line_height | 10 |
| 3.6.2.4 | num_page | 10 |
| 3.6.2.5 | pdf | 10 |
| 3.6.2.6 | pref | 10 |
| 3.6.2.7 | stat_print | 10 |
| 3.6.2.8 | table_line_height | 10 |
| 3.6.2.9 | total_points_ranking_print | 10 |
| 3.7 | export_pdf_preferences Struct Reference | 10 |
| 3.7.1 | Detailed Description | 11 |
| 3.7.2 | Field Documentation | 11 |
| 3.7.2.1 | charset | 11 |
| 3.7.2.2 | direction | 11 |
| 3.7.2.3 | font_size | 11 |
| 3.7.2.4 | margin | 11 |
| 3.7.2.5 | pdf_size_for_chart | 11 |
| 3.7.2.6 | ranking_turn | 11 |
| 3.7.2.7 | size | 11 |
| 3.7.2.8 | total_points_turn | 11 |

| | | |
|----------|-----------------------------------|----|
| 3.8 | game_config Struct Reference | 12 |
| 3.8.1 | Detailed Description | 12 |
| 3.8.2 | Field Documentation | 12 |
| 3.8.2.1 | begin_score | 12 |
| 3.8.2.2 | decimal_place | 12 |
| 3.8.2.3 | first_way | 12 |
| 3.8.2.4 | max | 12 |
| 3.8.2.5 | name | 12 |
| 3.8.2.6 | nb_max | 12 |
| 3.8.2.7 | turn_based | 12 |
| 3.8.2.8 | use_distributor | 13 |
| 3.9 | list_game_config Struct Reference | 13 |
| 3.9.1 | Detailed Description | 13 |
| 3.9.2 | Field Documentation | 13 |
| 3.9.2.1 | game_configs | 13 |
| 3.9.2.2 | name_game_config | 13 |
| 3.9.2.3 | nb_config | 13 |
| 3.10 | main_window_side Struct Reference | 13 |
| 3.10.1 | Detailed Description | 13 |
| 3.10.2 | Field Documentation | 14 |
| 3.10.2.1 | calculator | 14 |
| 3.10.2.2 | game_information | 14 |
| 3.10.2.3 | ranking | 14 |
| 3.11 | main_window_size Struct Reference | 14 |
| 3.11.1 | Detailed Description | 14 |
| 3.11.2 | Field Documentation | 14 |
| 3.11.2.1 | height | 14 |
| 3.11.2.2 | is_maximize | 14 |
| 3.11.2.3 | width | 14 |
| 3.12 | one_preferences Union Reference | 15 |
| 3.12.1 | Field Documentation | 15 |
| 3.12.1.1 | chart | 15 |
| 3.12.1.2 | diff | 15 |
| 3.12.1.3 | pdf | 15 |
| 3.12.1.4 | score | 15 |
| 3.12.1.5 | side | 15 |
| 3.12.1.6 | size | 15 |
| 3.12.1.7 | toolbar | 15 |
| 3.13 | preferences Struct Reference | 15 |
| 3.13.1 | Detailed Description | 15 |

| | | |
|-----------|--|----|
| 3.13.2 | Field Documentation | 16 |
| 3.13.2.1 | chart | 16 |
| 3.13.2.2 | diff | 16 |
| 3.13.2.3 | pdf | 16 |
| 3.13.2.4 | score | 16 |
| 3.13.2.5 | side | 16 |
| 3.13.2.6 | size | 16 |
| 3.13.2.7 | toolbar | 16 |
| 3.14 | score_display Struct Reference | 16 |
| 3.14.1 | Detailed Description | 16 |
| 3.14.2 | Field Documentation | 16 |
| 3.14.2.1 | edit_suppr | 16 |
| 3.14.2.2 | ranking | 16 |
| 3.14.2.3 | total_points | 16 |
| 3.15 | toolbar_button_preferences_struct Struct Reference | 17 |
| 3.15.1 | Detailed Description | 17 |
| 3.15.2 | Field Documentation | 17 |
| 3.15.2.1 | about | 17 |
| 3.15.2.2 | copy | 17 |
| 3.15.2.3 | cut | 17 |
| 3.15.2.4 | delete | 17 |
| 3.15.2.5 | delete_file | 18 |
| 3.15.2.6 | export | 18 |
| 3.15.2.7 | exportation_preferences | 18 |
| 3.15.2.8 | game_configuration_preferences | 18 |
| 3.15.2.9 | new | 18 |
| 3.15.2.10 | open | 18 |
| 3.15.2.11 | paste | 18 |
| 3.15.2.12 | preferences | 18 |
| 3.15.2.13 | properties | 18 |
| 3.15.2.14 | redo | 18 |
| 3.15.2.15 | save_as | 18 |
| 3.15.2.16 | separator_1 | 18 |
| 3.15.2.17 | separator_2 | 19 |
| 3.15.2.18 | separator_3 | 19 |
| 3.15.2.19 | separator_4 | 19 |
| 3.15.2.20 | separator_5 | 19 |
| 3.15.2.21 | separator_6 | 19 |
| 3.15.2.22 | toolbar_button_preferences | 19 |
| 3.15.2.23 | undo | 19 |

| | | |
|----------|------------------------------------|-----------|
| 4 | File Documentation | 21 |
| 4.1 | calculator.c File Reference | 21 |
| 4.1.1 | Detailed Description | 22 |
| 4.1.2 | Function Documentation | 22 |
| 4.1.2.1 | calculateFromString | 22 |
| 4.1.2.2 | calculatorListCalculate | 23 |
| 4.1.2.3 | calculatorListDelete | 23 |
| 4.1.2.4 | calculatorListDeleteCurrent | 24 |
| 4.1.2.5 | calculatorListEmpty | 25 |
| 4.1.2.6 | calculatorListFirst | 26 |
| 4.1.2.7 | calculatorListGetCurrentNumber | 27 |
| 4.1.2.8 | calculatorListGetCurrentOperator | 27 |
| 4.1.2.9 | calculatorListInit | 27 |
| 4.1.2.10 | calculatorListInsertAfterCurrent | 27 |
| 4.1.2.11 | calculatorListInsertBeforeCurrent | 28 |
| 4.1.2.12 | calculatorListInsertLast | 29 |
| 4.1.2.13 | calculatorListInsertLastFromString | 30 |
| 4.1.2.14 | calculatorListIsCurrentNumber | 30 |
| 4.1.2.15 | calculatorListLast | 30 |
| 4.1.2.16 | calculatorListNext | 30 |
| 4.1.2.17 | calculatorListOne | 31 |
| 4.1.2.18 | calculatorListOutOfList | 31 |
| 4.1.2.19 | calculatorListPrevious | 31 |
| 4.1.2.20 | calculatorListPrint | 31 |
| 4.1.2.21 | calculatorListSetOnFirst | 32 |
| 4.1.2.22 | calculatorListSetOnLast | 32 |
| 4.1.2.23 | calculatorSearchNextOperatorString | 32 |
| 4.1.2.24 | deleteCalculatorNodeList | 32 |
| 4.1.2.25 | newCalculatorNodeList | 33 |
| 4.2 | calculator.h File Reference | 33 |
| 4.2.1 | Detailed Description | 34 |
| 4.2.2 | Typedef Documentation | 34 |
| 4.2.2.1 | calculatorNodeList | 34 |
| 4.2.3 | Function Documentation | 34 |
| 4.2.3.1 | calculateFromString | 34 |
| 4.2.3.2 | calculatorListCalculate | 36 |
| 4.2.3.3 | calculatorListDelete | 36 |
| 4.2.3.4 | calculatorListDeleteCurrent | 37 |
| 4.2.3.5 | calculatorListEmpty | 38 |
| 4.2.3.6 | calculatorListGetCurrentNumber | 39 |

| | | |
|----------|------------------------------------|----|
| 4.2.3.7 | calculatorListGetCurrentOperator | 40 |
| 4.2.3.8 | calculatorListInit | 40 |
| 4.2.3.9 | calculatorListInsertAfterCurrent | 40 |
| 4.2.3.10 | calculatorListInsertBeforeCurrent | 41 |
| 4.2.3.11 | calculatorListInsertLast | 42 |
| 4.2.3.12 | calculatorListInsertLastFromString | 43 |
| 4.2.3.13 | calculatorListIsCurrentNumber | 43 |
| 4.2.3.14 | calculatorListLast | 43 |
| 4.2.3.15 | calculatorListFirst | 43 |
| 4.2.3.16 | calculatorListNext | 43 |
| 4.2.3.17 | calculatorListOne | 44 |
| 4.2.3.18 | calculatorListOutOfList | 44 |
| 4.2.3.19 | calculatorListPrevious | 44 |
| 4.2.3.20 | calculatorListPrint | 44 |
| 4.2.3.21 | calculatorListSetOnFirst | 45 |
| 4.2.3.22 | calculatorListSetOnLast | 45 |
| 4.2.3.23 | calculatorSearchNextOperatorString | 45 |
| 4.2.3.24 | deleteCalculatorNodeList | 45 |
| 4.2.3.25 | newCalculatorNodeList | 46 |
| 4.3 | csu_files.c File Reference | 46 |
| 4.3.1 | Detailed Description | 46 |
| 4.3.2 | Function Documentation | 47 |
| 4.3.2.1 | addXmlBoolNode | 47 |
| 4.3.2.2 | addXmlFloatNode | 47 |
| 4.3.2.3 | addXmlFloatNodeIntProp | 47 |
| 4.3.2.4 | addXmlIntNode | 48 |
| 4.3.2.5 | addXmlStringNode | 48 |
| 4.3.2.6 | openFileCsuExtension | 48 |
| 4.3.2.7 | readCsuFile | 49 |
| 4.3.2.8 | readCsuXmlFile | 49 |
| 4.3.2.9 | writeCsuFile | 50 |
| 4.3.2.10 | writeCsuXmlFile | 51 |
| 4.3.2.11 | writeFileNewTurn | 52 |
| 4.4 | csu_files.h File Reference | 52 |
| 4.4.1 | Detailed Description | 53 |
| 4.4.2 | Macro Definition Documentation | 53 |
| 4.4.2.1 | FILE_EXTENSION_CSU | 53 |
| 4.4.2.2 | SIZE_MAX_FILE_NAME | 53 |
| 4.4.2.3 | STRING_CHECK_CSU_FILE | 54 |
| 4.4.3 | Function Documentation | 54 |

| | | |
|----------|--|----|
| 4.4.3.1 | addXmlBoolNode | 54 |
| 4.4.3.2 | addXmlFloatNode | 54 |
| 4.4.3.3 | addXmlFloatNodeIntProp | 54 |
| 4.4.3.4 | addXmlIntNode | 55 |
| 4.4.3.5 | addXmlStringNode | 55 |
| 4.4.3.6 | openFileCsuExtension | 55 |
| 4.4.3.7 | readCsuFile | 56 |
| 4.4.3.8 | readCsuXmlFile | 56 |
| 4.4.3.9 | writeCsuFile | 57 |
| 4.4.3.10 | writeCsuXmlFile | 58 |
| 4.4.3.11 | writeFileNewTurn | 59 |
| 4.5 | csu_struct.c File Reference | 59 |
| 4.5.1 | Detailed Description | 60 |
| 4.5.2 | Function Documentation | 60 |
| 4.5.2.1 | addDistributorCsuStruct | 60 |
| 4.5.2.2 | changeDistributor | 61 |
| 4.5.2.3 | closeCsuStruct | 61 |
| 4.5.2.4 | copyCsuStruct | 61 |
| 4.5.2.5 | deleteTurn | 62 |
| 4.5.2.6 | differeentsPlayerName | 62 |
| 4.5.2.7 | endNewTurn | 63 |
| 4.5.2.8 | exceedMaxNumber | 64 |
| 4.5.2.9 | lastRankAtTurn | 64 |
| 4.5.2.10 | maxNbTurn | 65 |
| 4.5.2.11 | meanPoints | 65 |
| 4.5.2.12 | nbTurnBest | 65 |
| 4.5.2.13 | nbTurnFirst | 66 |
| 4.5.2.14 | nbTurnLast | 66 |
| 4.5.2.15 | nbTurnWorst | 67 |
| 4.5.2.16 | newCsuStruct | 67 |
| 4.5.2.17 | pointsAtTurn | 67 |
| 4.5.2.18 | rankAtTurn | 68 |
| 4.5.2.19 | rankCalculation | 68 |
| 4.5.2.20 | searchIndexFromPosition | 69 |
| 4.5.2.21 | searchPlayerIndex | 69 |
| 4.5.2.22 | startNewTurn | 70 |
| 4.6 | csu_struct.h File Reference | 70 |
| 4.6.1 | Detailed Description | 71 |
| 4.6.2 | Macro Definition Documentation | 71 |
| 4.6.2.1 | SIZE_MAX_NAME | 71 |

| | | |
|----------|--------------------------|----|
| 4.6.2.2 | VERSION | 71 |
| 4.6.3 | Function Documentation | 71 |
| 4.6.3.1 | addDistributorCsuStruct | 71 |
| 4.6.3.2 | changeDistributor | 72 |
| 4.6.3.3 | closeCsuStruct | 72 |
| 4.6.3.4 | copyCsuStruct | 72 |
| 4.6.3.5 | deleteTurn | 73 |
| 4.6.3.6 | differeentsPlayerName | 73 |
| 4.6.3.7 | endNewTurn | 74 |
| 4.6.3.8 | exceedMaxNumber | 75 |
| 4.6.3.9 | lastRankAtTurn | 75 |
| 4.6.3.10 | maxNbTurn | 76 |
| 4.6.3.11 | meanPoints | 76 |
| 4.6.3.12 | nbTurnBest | 76 |
| 4.6.3.13 | nbTurnFirst | 77 |
| 4.6.3.14 | nbTurnLast | 77 |
| 4.6.3.15 | nbTurnWorst | 78 |
| 4.6.3.16 | newCsuStruct | 78 |
| 4.6.3.17 | pointsAtTurn | 78 |
| 4.6.3.18 | rankAtTurn | 79 |
| 4.6.3.19 | rankCalculation | 79 |
| 4.6.3.20 | searchIndexFromPosition | 80 |
| 4.6.3.21 | searchPlayerIndex | 80 |
| 4.6.3.22 | startNewTurn | 81 |
| 4.7 | export.c File Reference | 81 |
| 4.7.1 | Detailed Description | 82 |
| 4.7.2 | Function Documentation | 82 |
| 4.7.2.1 | addPodiumPdf | 82 |
| 4.7.2.2 | addStatsPdf | 82 |
| 4.7.2.3 | addTotalPointsRankingPdf | 83 |
| 4.7.2.4 | canUseUtf8Pdf | 83 |
| 4.7.2.5 | closeExportPdf | 84 |
| 4.7.2.6 | createFirstPagePdf | 84 |
| 4.7.2.7 | createOtherPagePdf | 84 |
| 4.7.2.8 | createPdfGrid | 85 |
| 4.7.2.9 | errorHandler | 85 |
| 4.7.2.10 | exportToCsv | 86 |
| 4.7.2.11 | exportToM | 86 |
| 4.7.2.12 | exportToPdf | 87 |
| 4.7.2.13 | initializePdfExport | 88 |

| | | |
|----------|--|-----|
| 4.7.2.14 | pdfShowText | 88 |
| 4.7.2.15 | pdfTextOutTable | 89 |
| 4.7.2.16 | printLegendPdf | 89 |
| 4.7.2.17 | printNamesPdf | 90 |
| 4.7.2.18 | printPointsPdf | 90 |
| 4.7.2.19 | tableWidthCalculatePdf | 91 |
| 4.8 | export.h File Reference | 91 |
| 4.8.1 | Detailed Description | 92 |
| 4.8.2 | Macro Definition Documentation | 93 |
| 4.8.2.1 | DEFAULT_FONT_SIZE | 93 |
| 4.8.2.2 | DEFAULT_MARGIN | 93 |
| 4.8.2.3 | TEXT_BUFFER_SIZE | 93 |
| 4.8.3 | Enumeration Type Documentation | 93 |
| 4.8.3.1 | CharacterSetPdf | 93 |
| 4.8.4 | Function Documentation | 93 |
| 4.8.4.1 | addPodiumPdf | 93 |
| 4.8.4.2 | addStatsPdf | 93 |
| 4.8.4.3 | addTotalPointsRankingPdf | 94 |
| 4.8.4.4 | canUseUtf8Pdf | 94 |
| 4.8.4.5 | closeExportPdf | 95 |
| 4.8.4.6 | createFirstPagePdf | 96 |
| 4.8.4.7 | createOtherPagePdf | 96 |
| 4.8.4.8 | createPdfGrid | 97 |
| 4.8.4.9 | errorHandler | 97 |
| 4.8.4.10 | exportToCsv | 97 |
| 4.8.4.11 | exportToM | 98 |
| 4.8.4.12 | exportToPdf | 99 |
| 4.8.4.13 | initializePdfExport | 100 |
| 4.8.4.14 | pdfShowText | 100 |
| 4.8.4.15 | pdfTextOutTable | 101 |
| 4.8.4.16 | printLegendPdf | 101 |
| 4.8.4.17 | printNamesPdf | 102 |
| 4.8.4.18 | printPointsPdf | 102 |
| 4.8.4.19 | tableWidthCalculatePdf | 103 |
| 4.9 | file.c File Reference | 103 |
| 4.9.1 | Detailed Description | 104 |
| 4.9.2 | Function Documentation | 104 |
| 4.9.2.1 | closeFile | 104 |
| 4.9.2.2 | deleteFile | 104 |
| 4.9.2.3 | openFile | 105 |

| | | |
|-----------|-----------------------------------|-----|
| 4.9.2.4 | readFileSize | 105 |
| 4.9.2.5 | renameFile | 105 |
| 4.10 | file.h File Reference | 106 |
| 4.10.1 | Detailed Description | 106 |
| 4.10.2 | Function Documentation | 106 |
| 4.10.2.1 | closeFile | 106 |
| 4.10.2.2 | deleteFile | 107 |
| 4.10.2.3 | openFile | 107 |
| 4.10.2.4 | readFileSize | 108 |
| 4.10.2.5 | renameFile | 108 |
| 4.11 | filename.c File Reference | 109 |
| 4.11.1 | Detailed Description | 109 |
| 4.11.2 | Function Documentation | 109 |
| 4.11.2.1 | addFileCsuExtension | 109 |
| 4.11.2.2 | addFileCsvExtension | 110 |
| 4.11.2.3 | addFileDatExtension | 110 |
| 4.11.2.4 | addFileExtension | 110 |
| 4.11.2.5 | addFileGnuplotExtension | 111 |
| 4.11.2.6 | addFilePdfExtension | 111 |
| 4.11.2.7 | checkFilename | 111 |
| 4.11.2.8 | checkPath | 112 |
| 4.11.2.9 | getFolderFromFilename | 112 |
| 4.11.2.10 | getSimpleFilenameFromFullFilename | 113 |
| 4.11.2.11 | readHomePath | 113 |
| 4.11.2.12 | readHomePathSlash | 113 |
| 4.11.2.13 | removeFileExtension | 113 |
| 4.11.2.14 | removeFilenameExtension | 113 |
| 4.12 | filename.h File Reference | 114 |
| 4.12.1 | Detailed Description | 114 |
| 4.12.2 | Function Documentation | 114 |
| 4.12.2.1 | addFileCsuExtension | 114 |
| 4.12.2.2 | addFileCsvExtension | 115 |
| 4.12.2.3 | addFileDatExtension | 115 |
| 4.12.2.4 | addFileExtension | 115 |
| 4.12.2.5 | addFileGnuplotExtension | 116 |
| 4.12.2.6 | addFilePdfExtension | 116 |
| 4.12.2.7 | checkFilename | 116 |
| 4.12.2.8 | checkPath | 117 |
| 4.12.2.9 | getFolderFromFilename | 117 |
| 4.12.2.10 | getSimpleFilenameFromFullFilename | 118 |

| | |
|--|-----|
| 4.12.2.11 readHomePath | 118 |
| 4.12.2.12 readHomePathSlash | 118 |
| 4.12.2.13 removeFileExtension | 118 |
| 4.12.2.14 removeFilenameExtension | 118 |
| 4.13 game_config.c File Reference | 119 |
| 4.13.1 Detailed Description | 119 |
| 4.13.2 Function Documentation | 119 |
| 4.13.2.1 addConfigListFile | 119 |
| 4.13.2.2 addConfigListGameConfig | 120 |
| 4.13.2.3 closeListGameConfig | 120 |
| 4.13.2.4 exportConfigFile | 120 |
| 4.13.2.5 importConfigFile | 121 |
| 4.13.2.6 makeConfigListFile | 121 |
| 4.13.2.7 newConfigFile | 122 |
| 4.13.2.8 newListGameConfig | 122 |
| 4.13.2.9 newListGameConfigFromImport | 123 |
| 4.13.2.10 readConfigFile | 123 |
| 4.13.2.11 readConfigListFile | 124 |
| 4.13.2.12 readXmlListGameConfig | 124 |
| 4.13.2.13 readXmlListGameConfigWithId | 125 |
| 4.13.2.14 removeConfigFile | 125 |
| 4.13.2.15 removeConfigListFile | 125 |
| 4.13.2.16 removeConfigListGameConfig | 125 |
| 4.13.2.17 writeXmlListGameConfig | 126 |
| 4.13.2.18 writeXmlListGameConfigWithId | 126 |
| 4.14 game_config.h File Reference | 126 |
| 4.14.1 Detailed Description | 127 |
| 4.14.2 Macro Definition Documentation | 128 |
| 4.14.2.1 CONFIGURATION_FILE_NAME | 128 |
| 4.14.2.2 CONFIGURATION_FOLDER_NAME | 128 |
| 4.14.2.3 CONFIGURATION_XML_FILENAME | 128 |
| 4.14.2.4 GAME_CONFIG_FILE_XML_VERSION | 128 |
| 4.14.2.5 STRING_CHECK_GAME_CONFIG | 128 |
| 4.14.3 Function Documentation | 128 |
| 4.14.3.1 addConfigListFile | 128 |
| 4.14.3.2 addConfigListGameConfig | 128 |
| 4.14.3.3 closeListGameConfig | 129 |
| 4.14.3.4 exportConfigFile | 129 |
| 4.14.3.5 importConfigFile | 129 |
| 4.14.3.6 makeConfigListFile | 130 |

| | | |
|-----------|--|-----|
| 4.14.3.7 | newConfigFile | 131 |
| 4.14.3.8 | newListGameConfig | 132 |
| 4.14.3.9 | newListGameConfigFromImport | 132 |
| 4.14.3.10 | readConfigFile | 133 |
| 4.14.3.11 | readConfigListFile | 133 |
| 4.14.3.12 | readXmlListGameConfig | 134 |
| 4.14.3.13 | readXmlListGameConfigWithId | 134 |
| 4.14.3.14 | removeConfigFile | 135 |
| 4.14.3.15 | removeConfigListFile | 136 |
| 4.14.3.16 | removeConfigListGameConfig | 136 |
| 4.14.3.17 | writeXmlListGameConfig | 137 |
| 4.14.3.18 | writeXmlListGameConfigWithId | 137 |
| 4.15 | gnuplot.c File Reference | 137 |
| 4.15.1 | Detailed Description | 137 |
| 4.15.2 | Function Documentation | 138 |
| 4.15.2.1 | exportToGnuplotData | 138 |
| 4.15.2.2 | exportToGnuplotFile | 138 |
| 4.15.2.3 | exportToGnuplotScript | 139 |
| 4.16 | gnuplot.h File Reference | 140 |
| 4.16.1 | Detailed Description | 140 |
| 4.16.2 | Function Documentation | 140 |
| 4.16.2.1 | exportToGnuplotData | 140 |
| 4.16.2.2 | exportToGnuplotFile | 141 |
| 4.16.2.3 | exportToGnuplotScript | 142 |
| 4.17 | libcsuper.h File Reference | 142 |
| 4.17.1 | Detailed Description | 142 |
| 4.17.2 | Macro Definition Documentation | 143 |
| 4.17.2.1 | NOT_LIBCSUPER | 143 |
| 4.18 | main_argument.c File Reference | 143 |
| 4.18.1 | Detailed Description | 143 |
| 4.18.2 | Function Documentation | 143 |
| 4.18.2.1 | displayHelp | 143 |
| 4.18.2.2 | searchArgument | 144 |
| 4.19 | main_argument.h File Reference | 144 |
| 4.19.1 | Detailed Description | 145 |
| 4.19.2 | Macro Definition Documentation | 145 |
| 4.19.2.1 | STRING_EXPORT_TO_CSV | 145 |
| 4.19.2.2 | STRING_EXPORT_TO_CSV_RED | 145 |
| 4.19.2.3 | STRING_EXPORT_TO_GNUPLOT | 145 |
| 4.19.2.4 | STRING_EXPORT_TO_GNUPLOT_RED | 145 |

| | | |
|-----------|---|-----|
| 4.19.2.5 | STRING_EXPORT_TO_M | 146 |
| 4.19.2.6 | STRING_EXPORT_TO_M_RED | 146 |
| 4.19.2.7 | STRING_EXPORT_TO_PDF | 146 |
| 4.19.2.8 | STRING_EXPORT_TO_PDF_RED | 146 |
| 4.19.2.9 | STRING_HELP | 146 |
| 4.19.2.10 | STRING_HELP_RED | 146 |
| 4.19.2.11 | STRING_OPEN_FILE | 146 |
| 4.19.2.12 | STRING_OPEN_FILE_RED | 146 |
| 4.19.2.13 | STRING_READ_FILE | 146 |
| 4.19.2.14 | STRING_READ_FILE_RED | 146 |
| 4.19.3 | Enumeration Type Documentation | 146 |
| 4.19.3.1 | main_argument_function | 146 |
| 4.19.4 | Function Documentation | 147 |
| 4.19.4.1 | displayHelp | 147 |
| 4.19.4.2 | searchArgument | 147 |
| 4.20 | preferences_files.c File Reference | 147 |
| 4.20.1 | Detailed Description | 148 |
| 4.20.2 | Function Documentation | 148 |
| 4.20.2.1 | changeSystemPath | 148 |
| 4.20.2.2 | createFileChartExportation | 149 |
| 4.20.2.3 | createFileDifferenceBetweenPlayer | 149 |
| 4.20.2.4 | createFileMainWidowSize | 150 |
| 4.20.2.5 | createFileMainWindowSide | 150 |
| 4.20.2.6 | createFilePdfPreferences | 151 |
| 4.20.2.7 | createFileScoreDisplay | 151 |
| 4.20.2.8 | createFileSystemPath | 152 |
| 4.20.2.9 | createFileToolbarButtonPreferences | 152 |
| 4.20.2.10 | createPreferencesFolder | 153 |
| 4.20.2.11 | differeentsChartExportationStruct | 153 |
| 4.20.2.12 | differeentsTExportPdfPreferencesStruct | 153 |
| 4.20.2.13 | differeentsToolbarButtonPreferencesStruct | 154 |
| 4.20.2.14 | readFileChartExportation | 154 |
| 4.20.2.15 | readFileDifferenceBetweenPlayer | 154 |
| 4.20.2.16 | readFileMainWidowSize | 155 |
| 4.20.2.17 | readFileMainWindowSide | 155 |
| 4.20.2.18 | readFilePdfPreferences | 156 |
| 4.20.2.19 | readFileScoreDisplay | 156 |
| 4.20.2.20 | readFileSystemPath | 157 |
| 4.20.2.21 | readFileToolbarButtonPreferences | 157 |
| 4.20.2.22 | readSystemPath | 158 |

| | |
|--|-----|
| 4.20.2.23 readXmlPreferencesFile | 158 |
| 4.20.2.24 readXmlPreferencesFileType | 159 |
| 4.20.2.25 writeXmlPreferencesFile | 159 |
| 4.20.2.26 writeXmlPreferencesFileType | 160 |
| 4.21 preferences_files.h File Reference | 161 |
| 4.21.1 Detailed Description | 162 |
| 4.21.2 Macro Definition Documentation | 162 |
| 4.21.2.1 FILENAME_DIFFERENCE_BETWEEN_PLAYER | 162 |
| 4.21.2.2 FILENAME_MAIN_WINDOW_SIDE | 162 |
| 4.21.2.3 FILENAME_MAIN_WINDOW_SIZE | 162 |
| 4.21.2.4 FILENAME_PDF_PREFERENCES | 162 |
| 4.21.2.5 FILENAME_PREFERENCES_XML | 162 |
| 4.21.2.6 FILENAME_SCORE_DISPLAY | 163 |
| 4.21.2.7 FILENAME_SYSTEM_PATH | 163 |
| 4.21.2.8 FILENAME_TOOLBAR_BUTTON_PREFERENCES | 163 |
| 4.21.2.9 PREFERENCES_FILE_XML_VERSION | 163 |
| 4.21.2.10 PREFERENCES_FOLDER_NAME | 163 |
| 4.21.3 Enumeration Type Documentation | 163 |
| 4.21.3.1 preferences_type | 163 |
| 4.21.4 Function Documentation | 163 |
| 4.21.4.1 changeSystemPath | 163 |
| 4.21.4.2 createFileChartExportation | 164 |
| 4.21.4.3 createFileDifferenceBetweenPlayer | 164 |
| 4.21.4.4 createFileMainWidowSize | 165 |
| 4.21.4.5 createFileMainWindowSide | 165 |
| 4.21.4.6 createFilePdfPreferences | 166 |
| 4.21.4.7 createFileScoreDisplay | 166 |
| 4.21.4.8 createFileSystemPath | 167 |
| 4.21.4.9 createFileToolbarButtonPreferences | 167 |
| 4.21.4.10 createPreferencesFolder | 168 |
| 4.21.4.11 differentsChartExportationStruct | 168 |
| 4.21.4.12 differentsTExportPdfPreferencesStruct | 168 |
| 4.21.4.13 differentsToolbarButtonPreferencesStruct | 169 |
| 4.21.4.14 readFileChartExportation | 169 |
| 4.21.4.15 readFileDifferenceBetweenPlayer | 169 |
| 4.21.4.16 readFileMainWidowSize | 170 |
| 4.21.4.17 readFileMainWindowSide | 170 |
| 4.21.4.18 readFilePdfPreferences | 171 |
| 4.21.4.19 readFileScoreDisplay | 171 |
| 4.21.4.20 readFileSystemPath | 172 |

| | | |
|-----------|--|-----|
| 4.21.4.21 | readFileToolbarButtonPreferences | 172 |
| 4.21.4.22 | readSystemPath | 173 |
| 4.21.4.23 | readXmlPreferencesFile | 173 |
| 4.21.4.24 | readXmlPreferencesFileType | 174 |
| 4.21.4.25 | writeXmlPreferencesFile | 174 |
| 4.21.4.26 | writeXmlPreferencesFileType | 175 |
| 4.22 | share.c File Reference | 176 |
| 4.22.1 | Detailed Description | 176 |
| 4.22.2 | Function Documentation | 176 |
| 4.22.2.1 | clearScreen | 176 |
| 4.22.2.2 | compareFloatAscending | 177 |
| 4.22.2.3 | compareFloatDescending | 177 |
| 4.22.2.4 | convertFloatString | 177 |
| 4.22.2.5 | convertStringBool | 177 |
| 4.22.2.6 | convertStringFloat | 177 |
| 4.22.2.7 | convertStringInt | 178 |
| 4.22.2.8 | integerToYesNo | 178 |
| 4.22.2.9 | libcsuper_initialize | 178 |
| 4.22.2.10 | myAlloc | 178 |
| 4.22.2.11 | myRealloc | 179 |
| 4.22.2.12 | utf8ToLatin9 | 179 |
| 4.22.2.13 | wrongChoice | 179 |
| 4.23 | share.h File Reference | 180 |
| 4.23.1 | Detailed Description | 180 |
| 4.23.2 | Macro Definition Documentation | 181 |
| 4.23.2.1 | _ | 181 |
| 4.23.2.2 | CSUPER_VERSION | 181 |
| 4.23.3 | Function Documentation | 181 |
| 4.23.3.1 | clearScreen | 181 |
| 4.23.3.2 | compareFloatAscending | 181 |
| 4.23.3.3 | compareFloatDescending | 181 |
| 4.23.3.4 | convertFloatString | 182 |
| 4.23.3.5 | convertStringBool | 182 |
| 4.23.3.6 | convertStringFloat | 182 |
| 4.23.3.7 | convertStringInt | 182 |
| 4.23.3.8 | integerToYesNo | 183 |
| 4.23.3.9 | libcsuper_initialize | 183 |
| 4.23.3.10 | myAlloc | 183 |
| 4.23.3.11 | myRealloc | 184 |
| 4.23.3.12 | utf8ToLatin9 | 184 |

| | |
|---------------------------------|------------|
| 4.23.3.13 wrongChoice | 184 |
| Index | 185 |

Chapter 1

Data Structure Index

1.1 Data Structures

Here are the data structures with brief descriptions:

| | |
|--|----|
| calculatorList | |
| CalculatorList of tree nodes | 5 |
| calculatorNodeList | |
| Data element of a calculatorList , a number or an operator | 5 |
| chart_exportation | 6 |
| csuStruct | 7 |
| difference_between_player | 9 |
| export_pdf | 9 |
| export_pdf_preferences | 10 |
| game_config | 12 |
| list_game_config | 13 |
| main_window_side | 13 |
| main_window_size | 14 |
| one_preferences | 15 |
| preferences | 15 |
| score_display | 16 |
| toolbar_button_preferences_struct | 17 |

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

| | |
|--|-----|
| calculator.c | 21 |
| calculator.h | 33 |
| csu_files.c | |
| Files management | 46 |
| csu_files.h | |
| Files management | 52 |
| csu_struct.c | |
| Management of the csu files | 59 |
| csu_struct.h | |
| Management of the csu files header | 70 |
| export.c | |
| Export function | 81 |
| export.h | |
| Header for the export function | 91 |
| file.c | |
| Files function of libcsuper | 103 |
| file.h | |
| Header for the files function of libcsuper | 106 |
| filename.c | |
| Essential function of libcsuper | 109 |
| filename.h | |
| Header for the essential function of libcsuper | 114 |
| game_config.c | |
| Game configuration | 119 |
| game_config.h | |
| Game configurations | 126 |
| gnuplot.c | 137 |
| gnuplot.h | 140 |
| libcsuper.h | |
| Inclusion of all header files of libcsuper | 142 |
| main_argument.c | |
| Begin csuper | 143 |
| main_argument.h | |
| Begin csuper | 144 |
| preferences_files.c | |
| Function which store preferences into files | 147 |
| preferences_files.h | |
| Prototypes des fonctions qui l'emrankment des fichiers sauvegardes | 161 |

| | | |
|-------------------------|--|-----|
| share.c | Essential function of libcsuper | 176 |
| share.h | Header for the essential function of libcsuper | 180 |

Chapter 3

Data Structure Documentation

3.1 calculatorList Struct Reference

calculatorList of tree nodes

```
#include <calculator.h>
```

Data Fields

- calculatorNodeList * first
- calculatorNodeList * current
- calculatorNodeList * last

3.1.1 Detailed Description

calculatorList of tree nodes

3.1.2 Field Documentation

3.1.2.1 calculatorNodeList* current

A pointer to the current calculatorList element

3.1.2.2 calculatorNodeList* first

A pointer to the first calculatorList element

3.1.2.3 calculatorNodeList* last

A pointer to the last calculatorList element

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

- calculator.h

3.2 calculatorNodeList Struct Reference

Data element of a calculatorList, a number or an operator.

```
#include <calculator.h>
```

Data Fields

- bool [is_number](#)
- double [number](#)
- char [operator](#)
- struct [calculatorNodeList](#) * [next](#)
- struct [calculatorNodeList](#) * [previous](#)

3.2.1 Detailed Description

Data element of a [calculatorList](#), a number or an operator.

3.2.2 Field Documentation

3.2.2.1 bool [is_number](#)

Defined if this is a number or not

3.2.2.2 struct [calculatorNodeList](#)* [next](#)

A pointer to the next element in the [calculatorList](#)

3.2.2.3 double [number](#)

The number value

3.2.2.4 char [operator](#)

The operator

3.2.2.5 struct [calculatorNodeList](#)* [previous](#)

A pointer to the previous element in the [calculatorList](#)

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

- [calculator.h](#)

3.3 [chart_exportation](#) Struct Reference

```
#include <preferences_files.h>
```

Data Fields

- int [width](#)
- int [height](#)
- bool [total_points](#)

3.3.1 Detailed Description

Indicate what will be display in the left side of the main window

3.3.2 Field Documentation

3.3.2.1 int height

The height of the chart

3.3.2.2 bool total_points

Total points or points

3.3.2.3 int width

The width of the chart

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

- [preferences_files.h](#)

3.4 csuStruct Struct Reference

```
#include <csu_struct.h>
```

Data Fields

- float [version](#)
- float [size_max_name](#)
- float [day](#)
- float [month](#)
- float [year](#)
- float [nb_player](#)
- [game_config](#) config
- char ** [player_names](#)
- float * [total_points](#)
- float * [rank](#)
- float * [nb_turn](#)
- float [distributor](#)
- float ** [point](#)

3.4.1 Detailed Description

Represent a csu file

3.4.2 Field Documentation

3.4.2.1 game_config config

The game configuration.

3.4.2.2 float day

Day of the structure creation.

3.4.2.3 float distributor

Index of the distributor.

3.4.2.4 float month

Month of the structure creation.

3.4.2.5 float nb_player

Number of player.

3.4.2.6 float* nb_turn

Array containing the number of turn of all players.

3.4.2.7 char player_names**

Array containing the name of all players.

3.4.2.8 float point**

Array containing the points of all players in each turn.

3.4.2.9 float* rank

Array containing the rank of all players.

3.4.2.10 float size_max_name

Maximum size that can reach a player name.

3.4.2.11 float* total_points

Array containing the total score of all players.

3.4.2.12 float version

Version of the structure.

3.4.2.13 float year

Year of the structure creation.

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

- [csu_struct.h](#)

3.5 difference_between_player Struct Reference

```
#include <preferences_files.h>
```

Data Fields

- bool [consecutive](#)
- bool [first](#)
- bool [last](#)

3.5.1 Detailed Description

Indicate which difference between player will be displayed in the ranking

3.5.2 Field Documentation

3.5.2.1 bool consecutive

Between two player consecutive

3.5.2.2 bool first

Between the player and the first

3.5.2.3 bool last

Between the player and the last one

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

- [preferences_files.h](#)

3.6 export_pdf Struct Reference

```
#include <export.h>
```

Data Fields

- HPDF_Doc [pdf](#)
- HPDF_Font [font](#)
- int [line](#)
- int [num_page](#)
- int [line_height](#)
- int [table_line_height](#)
- [export_pdf_preferences](#) [pref](#)
- bool [total_points_ranking_print](#)
- bool [stat_print](#)

3.6.1 Detailed Description

Use to export a csu file to a pdf file

3.6.2 Field Documentation

3.6.2.1 HPDF_Font font

The font

3.6.2.2 int line

The number of the line to be display

3.6.2.3 int line_height

The line height

3.6.2.4 int num_page

The number of the page to be display

3.6.2.5 HPDF_Doc pdf

The pdf document

3.6.2.6 export_pdf_preferences pref

The user preferences

3.6.2.7 bool stat_print

Indicate if the stats are printed

3.6.2.8 int table_line_height

The line height of a table

3.6.2.9 bool total_points_ranking_print

Indicate if the total points and the ranking is printed

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

- [export.h](#)

3.7 export_pdf_preferences Struct Reference

```
#include <export.h>
```

Data Fields

- int [font_size](#)
- HPDF_PageSizes [size](#)
- HPDF_PageDirection [direction](#)
- [CharacterSetPdf](#) [charset](#)
- int [margin](#)
- bool [total_points_turn](#)
- bool [ranking_turn](#)
- bool [pdf_size_for_chart](#)

3.7.1 Detailed Description

Define the user preferences of the pdf

3.7.2 Field Documentation

3.7.2.1 [CharacterSetPdf](#) [charset](#)

The character set of the pdf document

3.7.2.2 [HPDF_PageDirection](#) [direction](#)

The direction of the pdf document

3.7.2.3 [int](#) [font_size](#)

The classical font size

3.7.2.4 [int](#) [margin](#)

The margin of the pdf document

3.7.2.5 [bool](#) [pdf_size_for_chart](#)

Indicate if we use the pdf size for the chart

3.7.2.6 [bool](#) [ranking_turn](#)

Indicate if we show the ranking in each turn

3.7.2.7 [HPDF_PageSizes](#) [size](#)

The size of the pdf document

3.7.2.8 [bool](#) [total_points_turn](#)

Indicate if we show the total points in each turn

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

- [export.h](#)

3.8 game_config Struct Reference

```
#include <csu_struct.h>
```

Data Fields

- float [nb_max](#)
- char [first_way](#)
- char [turn_based](#)
- char [use_distributor](#)
- char [decimal_place](#)
- char [max](#)
- char [name](#) [[SIZE_MAX_NAME](#)]
- float [begin_score](#)

3.8.1 Detailed Description

Represent a game configuration

3.8.2 Field Documentation

3.8.2.1 float begin_score

The score of all players in the beginning of the game

3.8.2.2 char decimal_place

The number of decimal place which are display

3.8.2.3 char first_way

Is 1 if the first is those has the maximum of points, -1 otherwise

3.8.2.4 char max

Is 1 if the game use a maximum, 0 if it's a minimum

3.8.2.5 char name[SIZE_MAX_NAME]

The name of the game configuration

3.8.2.6 float nb_max

Number maximum or minimum that can reach a player.

3.8.2.7 char turn_based

Is 1 if this is a turn-based game, 0 otherwise

3.8.2.8 char use_distributor

Is 1 if the game use a distributor, 0 otherwise

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

- [csu_struct.h](#)

3.9 list_game_config Struct Reference

```
#include <game_config.h>
```

Data Fields

- int [nb_config](#)
- char ** [name_game_config](#)
- [game_config](#) * [game_configs](#)

3.9.1 Detailed Description

Represent a list of game configuration

3.9.2 Field Documentation

3.9.2.1 game_config* game_configs

3.9.2.2 char** name_game_config

The list of the game configuration.

3.9.2.3 int nb_config

Number of game configuration.

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

- [game_config.h](#)

3.10 main_window_side Struct Reference

```
#include <preferences_files.h>
```

Data Fields

- bool [ranking](#)
- bool [calculator](#)
- bool [game_information](#)

3.10.1 Detailed Description

Indicate what will be display in the left side of the main window

3.10.2 Field Documentation

3.10.2.1 bool calculator

Display the calculator

3.10.2.2 bool game_information

Display the game information

3.10.2.3 bool ranking

Display the ranking

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

- [preferences_files.h](#)

3.11 main_window_size Struct Reference

```
#include <preferences_files.h>
```

Data Fields

- int [width](#)
- int [height](#)
- int [is_maximize](#)

3.11.1 Detailed Description

All component of the man window size

3.11.2 Field Documentation

3.11.2.1 int height

The height of the main window

3.11.2.2 int is_maximize

Said if the main window is maximize or not

3.11.2.3 int width

The width of the main window

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

- [preferences_files.h](#)

3.12 one_preferences Union Reference

```
#include <preferences_files.h>
```

Data Fields

- [toolbar_button_preferences_struct](#) toolbar
- [main_window_size](#) size
- [difference_between_player](#) diff
- [score_display](#) score
- [main_window_side](#) side
- [export_pdf_preferences](#) pdf
- [chart_exportation](#) chart

3.12.1 Field Documentation

3.12.1.1 [chart_exportation](#) chart

3.12.1.2 [difference_between_player](#) diff

3.12.1.3 [export_pdf_preferences](#) pdf

3.12.1.4 [score_display](#) score

3.12.1.5 [main_window_side](#) side

3.12.1.6 [main_window_size](#) size

3.12.1.7 [toolbar_button_preferences_struct](#) toolbar

The documentation for this union was generated from the following file:

- [preferences_files.h](#)

3.13 preferences Struct Reference

```
#include <preferences_files.h>
```

Data Fields

- [toolbar_button_preferences_struct](#) toolbar
- [main_window_size](#) size
- [difference_between_player](#) diff
- [score_display](#) score
- [main_window_side](#) side
- [export_pdf_preferences](#) pdf
- [chart_exportation](#) chart

3.13.1 Detailed Description

The preference structure

3.13.2 Field Documentation

3.13.2.1 `chart_exportation` chart

3.13.2.2 `difference_between_player` diff

3.13.2.3 `export_pdf_preferences` pdf

3.13.2.4 `score_display` score

3.13.2.5 `main_window_side` side

3.13.2.6 `main_window_size` size

3.13.2.7 `toolbar_button_preferences_struct` toolbar

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

- [preferences_files.h](#)

3.14 `score_display` Struct Reference

```
#include <preferences_files.h>
```

Data Fields

- bool `total_points`
- bool `ranking`
- bool `edit_suppr`

3.14.1 Detailed Description

Indicate if the total points and the ranking will be display in each turn

3.14.2 Field Documentation

3.14.2.1 `bool edit_suppr`

Display the edit and delete turn in each turn

3.14.2.2 `bool ranking`

Display the ranking in each turn

3.14.2.3 `bool total_points`

Display the total points in each turn

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

- [preferences_files.h](#)

3.15 toolbar_button_preferences_struct Struct Reference

```
#include <preferences_files.h>
```

Data Fields

- int [new](#)
- int [open](#)
- int [save_as](#)
- int [export](#)
- int [separator_6](#)
- int [delete_file](#)
- int [separator_1](#)
- int [undo](#)
- int [redo](#)
- int [separator_2](#)
- int [cut](#)
- int [copy](#)
- int [paste](#)
- int [delete](#)
- int [separator_3](#)
- int [properties](#)
- int [separator_4](#)
- int [preferences](#)
- int [game_configuration_preferences](#)
- int [toolbar_button_preferences](#)
- int [exportation_preferences](#)
- int [separator_5](#)
- int [about](#)

3.15.1 Detailed Description

Represent the toolbar button preferences

3.15.2 Field Documentation

3.15.2.1 int about

The about button

3.15.2.2 int copy

The copy button

3.15.2.3 int cut

The cut button

3.15.2.4 int delete

The delete button

3.15.2.5 int delete_file

The delete file button

3.15.2.6 int export

The export button

3.15.2.7 int exportation_preferences

The exportation preferences button

3.15.2.8 int game_configuration_preferences

The game configuration preferences button

3.15.2.9 int new

The new button

3.15.2.10 int open

The open button

3.15.2.11 int paste

The paste button

3.15.2.12 int preferences

The preferences button

3.15.2.13 int properties

The properties button

3.15.2.14 int redo

The redo button

3.15.2.15 int save_as

The save_as button

3.15.2.16 int separator_1

The separator 1

3.15.2.17 int separator_2

The separator 2

3.15.2.18 int separator_3

The separator 3

3.15.2.19 int separator_4

The separator 4

3.15.2.20 int separator_5

The separator 5

3.15.2.21 int separator_6

The separator 6

3.15.2.22 int toolbar_button_preferences

The toolbar button preferences button

3.15.2.23 int undo

The undo button

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

- [preferences_files.h](#)

Chapter 4

File Documentation

4.1 calculator.c File Reference

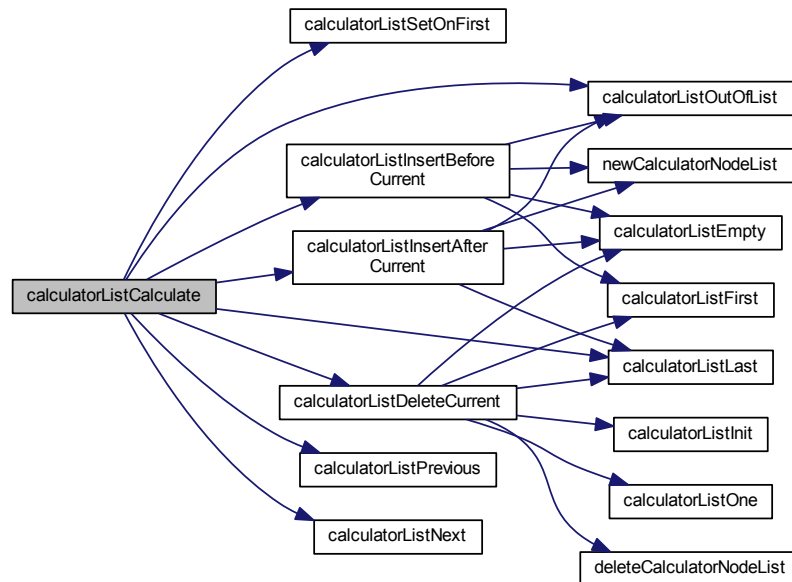
```
#include "calculator.h"
```

Functions

- [calculatorNodeList * newCalculatorNodeList](#) (bool is_number, double number, char operator, [calculatorNodeList *previous_node](#), [calculatorNodeList *next_node](#))
- void [deleteCalculatorNodeList](#) ([calculatorNodeList *node](#))
- void [calculatorListInit](#) ([calculatorList *list](#))
- bool [calculatorListEmpty](#) ([calculatorList *list](#))
- bool [calculatorListFirst](#) ([calculatorList *list](#))
- bool [calculatorListLast](#) ([calculatorList *list](#))
- bool [calculatorListOne](#) ([calculatorList *list](#))
- bool [calculatorListOutOfList](#) ([calculatorList *list](#))
- void [calculatorListSetOnFirst](#) ([calculatorList *list](#))
- void [calculatorListSetOnLast](#) ([calculatorList *list](#))
- void [calculatorListNext](#) ([calculatorList *list](#))
- void [calculatorListPrevious](#) ([calculatorList *list](#))
- double [calculatorListGetCurrentNumber](#) ([calculatorList *list](#))
- char [calculatorListGetCurrentOperator](#) ([calculatorList *list](#))
- bool [calculatorListIsCurrentNumber](#) ([calculatorList *list](#))
- void [calculatorListDelete](#) ([calculatorList *list](#))
- void [calculatorListPrint](#) ([calculatorList *list](#))
- bool [calculatorListInsertAfterCurrent](#) ([calculatorList *list](#), bool is_number, double number, char operator)
- bool [calculatorListInsertBeforeCurrent](#) ([calculatorList *list](#), bool is_number, double number, char operator)
- bool [calculatorListInsertLast](#) ([calculatorList *list](#), bool is_number, double number, char operator)
- bool [calculatorListDeleteCurrent](#) ([calculatorList *list](#))
- bool [calculatorListInsertLastFromString](#) (char *string, [calculatorList *list](#))
- int [calculatorSearchNextOperatorString](#) (char *string)
- bool [calculatorListCalculate](#) ([calculatorList *list](#), char operator1, char operator2)
- double [calculateFromString](#) (char *string)

4.1.2.2 bool calculatorListCalculate (calculatorList * list, char operator1, char operator2)

Here is the call graph for this function:



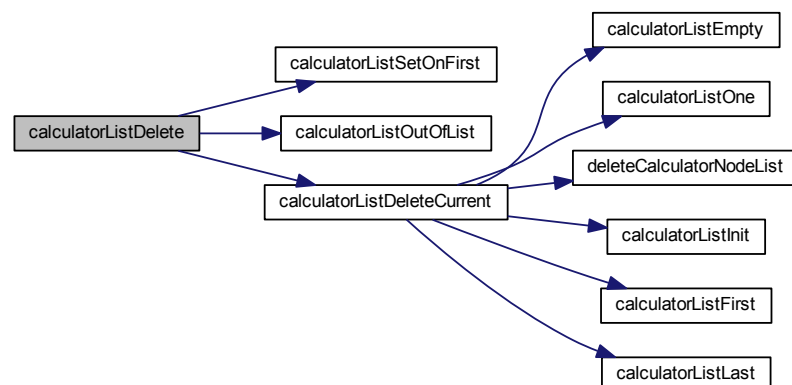
4.1.2.3 void calculatorListDelete (calculatorList * list)

Delete all the nodes of the list

Parameters

| | | |
|----|------|----------|
| in | list | the list |
|----|------|----------|

Here is the call graph for this function:



4.1.2.4 `bool calculatorListDeleteCurrent (calculatorList * list)`

Delete the current node to the list a set the current to the next or to the previous if the current was the last

Parameters

| | | |
|-----------|-------------|----------|
| <i>in</i> | <i>list</i> | the list |
|-----------|-------------|----------|

Returns

true if everything is OK, false otherwise

Calculate an expression with one of the two operator Scann all the list and if an operator is detected calculate the result and replace the expression to the result

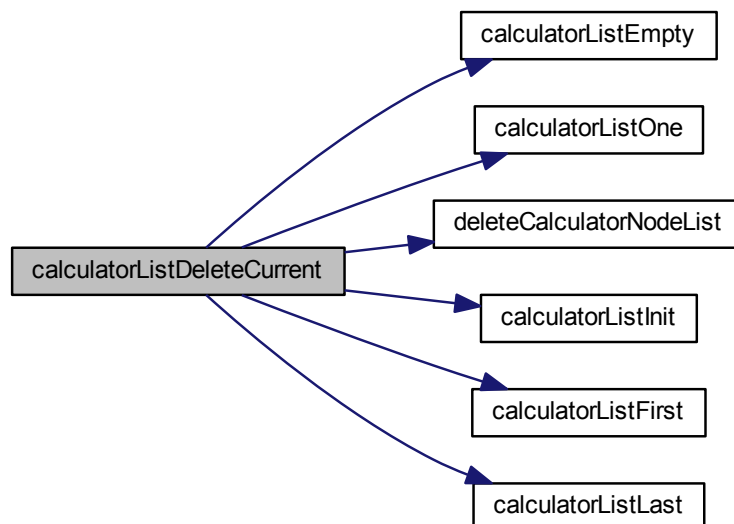
Parameters

| | | |
|-----------|-------------|----------|
| <i>in</i> | <i>list</i> | the list |
|-----------|-------------|----------|

Returns

true if everything is OK, false if there is no expression to evaluate

Here is the call graph for this function:



4.1.2.5 bool calculatorListEmpty (calculatorList * list)

Tests if it is an empty list.

Parameters

| | | |
|-----------|-------------|-----------------------|
| <i>in</i> | <i>list</i> | the list to be tested |
|-----------|-------------|-----------------------|

Returns

true if empty, false otherwise

4.1.2.6 `bool calculatorListFirst (calculatorList * list)`

Tests if the current node is the first one.

Parameters

| | | |
|-----------|-------------|-----------------------|
| <i>in</i> | <i>list</i> | the list to be tested |
|-----------|-------------|-----------------------|

Returns

true if the current node is the first one, false otherwise

4.1.2.7 double calculatorListGetCurrentNumber (calculatorList * *list*)

Return the number value stored in the current element.

Parameters

| | | |
|-----------|-------------|----------|
| <i>in</i> | <i>list</i> | the list |
|-----------|-------------|----------|

Returns

the number of the current element

4.1.2.8 char calculatorListGetCurrentOperator (calculatorList * *list*)

Return the operator value stored in the current element.

Parameters

| | | |
|-----------|-------------|----------|
| <i>in</i> | <i>list</i> | the list |
|-----------|-------------|----------|

Returns

the operator of the current element

4.1.2.9 void calculatorListInit (calculatorList * *list*)

Initialize the list structure members to be consistent with an empty list.

Parameters

| | | |
|------------|-------------|----------------------------|
| <i>out</i> | <i>list</i> | the list to be initialized |
|------------|-------------|----------------------------|

4.1.2.10 bool calculatorListInsertAfterCurrent (calculatorList * *list*, bool *is_number*, double *number*, char *operator*)

Insert a node after the current one

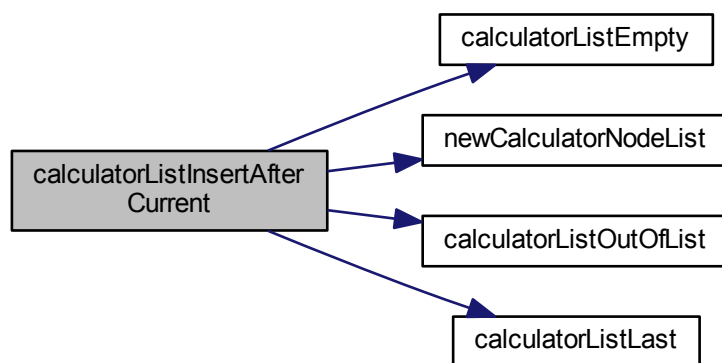
Parameters

| | | |
|-----------|------------------|----------------------------------|
| <i>in</i> | <i>is_number</i> | the new <i>is_number</i> element |
| <i>in</i> | <i>number</i> | the new number element |
| <i>in</i> | <i>operator</i> | the new operator element |
| <i>in</i> | <i>list</i> | the list |

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:



4.1.2.11 `bool calculatorListInsertBeforeCurrent (calculatorList * list, bool is_number, double number, char operator)`

Insert a node before the current one

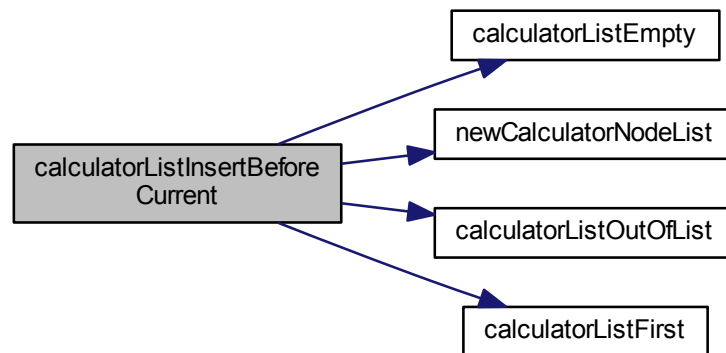
Parameters

| | | |
|----|------------------|---------------------------|
| in | <i>is_number</i> | the new is_number element |
| in | <i>number</i> | the new number element |
| in | <i>operator</i> | the new operator element |
| in | <i>list</i> | the list |

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:



4.1.2.12 `bool calculatorListInsertLast (calculatorList * list, bool is_number, double number, char operator)`

Insert a node in the last position of the list

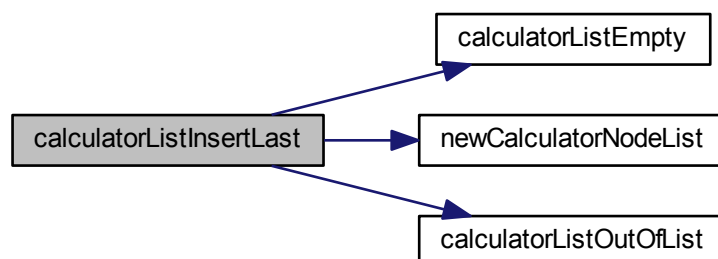
Parameters

| | | |
|----|------------------|---------------------------|
| in | <i>is_number</i> | the new is_number element |
| in | <i>number</i> | the new number element |
| in | <i>operator</i> | the new operator element |
| in | <i>list</i> | the list |

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:



4.1.2.13 bool calculatorListInsertLastFromString (char * *string*, calculatorList * *list*)

Convert the string into the list

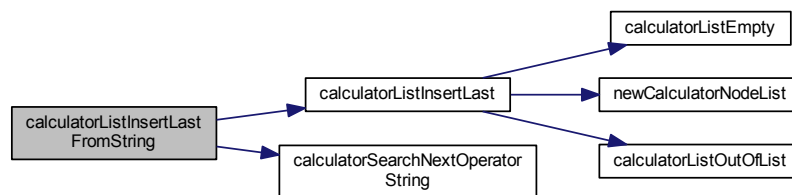
Parameters

| | | |
|----|---------------|--|
| in | <i>list</i> | the list |
| in | <i>string</i> | the string which contain all the element of the list (a mathematical expression) |

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:



4.1.2.14 bool calculatorListIsCurrentNumber (calculatorList * *list*)

Return the `ir_number` value stored in the current element.

Parameters

| | | |
|----|-------------|----------|
| in | <i>list</i> | the list |
|----|-------------|----------|

Returns

the `is_number` of the current element

4.1.2.15 bool calculatorListLast (calculatorList * *list*)

Tests if the current node is the last one.

Parameters

| | | |
|----|-------------|-----------------------|
| in | <i>list</i> | the list to be tested |
|----|-------------|-----------------------|

Returns

true if the current node is the last one, false otherwise

4.1.2.16 void calculatorListNext (calculatorList * *list*)

Set the current node on the next one.

Parameters

| | | |
|-----------|-------------|-------------------------|
| <i>in</i> | <i>list</i> | the list to be modified |
|-----------|-------------|-------------------------|

4.1.2.17 bool calculatorListOne (calculatorList * *list*)

Tests if there is only one element in the list

Parameters

| | | |
|-----------|-------------|-----------------------|
| <i>in</i> | <i>list</i> | the list to be tested |
|-----------|-------------|-----------------------|

Returns

true if there is only one element in the list, false otherwise

4.1.2.18 calculatorListOutOfList (calculatorList * *list*)

Tests if the current node is not valid (ie NULL).

Parameters

| | | |
|-----------|-------------|-----------------------|
| <i>in</i> | <i>list</i> | the list to be tested |
|-----------|-------------|-----------------------|

Returns

true if the current node is not valid, false otherwise

4.1.2.19 void calculatorListPrevious (calculatorList * *list*)

Set the current node on the previous one.

Parameters

| | | |
|-----------|-------------|-------------------------|
| <i>in</i> | <i>list</i> | the list to be modified |
|-----------|-------------|-------------------------|

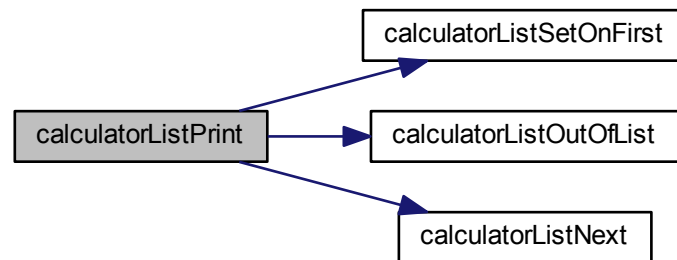
4.1.2.20 void calculatorListPrint (calculatorList * *list*)

Print the content of the list.

Parameters

| | | |
|-----------|-------------|--------------------------|
| <i>in</i> | <i>list</i> | the list to be displayed |
|-----------|-------------|--------------------------|

Here is the call graph for this function:



4.1.2.21 void calculatorListSetOnFirst (calculatorList * list)

Set the current node on the first one.

Parameters

| | | |
|-----------|-------------|-------------------------|
| <i>in</i> | <i>list</i> | the list to be modified |
|-----------|-------------|-------------------------|

4.1.2.22 void calculatorListSetOnLast (calculatorList * list)

Set the current node on the last one.

Parameters

| | | |
|-----------|-------------|-------------------------|
| <i>in</i> | <i>list</i> | the list to be modified |
|-----------|-------------|-------------------------|

4.1.2.23 unsigned int calculatorSearchNextOperatorString (char * string)

Search the next occurrence of an operator in a string

Parameters

| | | |
|-----------|---------------|--|
| <i>in</i> | <i>string</i> | the string which contain all the element of the list (a mathematical expression) |
|-----------|---------------|--|

Returns

the position of the next operator, 0 if there is no

4.1.2.24 void deleteCalculatorNodeList (calculatorNodeList * node)

Free a node

Parameters

| | | |
|-----|------|----------------------|
| out | node | the node to be freed |
|-----|------|----------------------|

4.1.2.25 **calculatorNodeList * newCalculatorNodeList** (bool *is_number*, double *number*, char *operator*, calculatorNodeList * *previous_node*, calculatorNodeList * *next_node*)

Allocate memory for the data element and initialize the structure members.

Parameters

| | | |
|----|----------------------|-------------------------------|
| in | <i>is_number</i> | the new is_number element |
| in | <i>number</i> | the new number element |
| in | <i>operator</i> | the new operator element |
| in | <i>previous_node</i> | the new previous_node element |
| in | <i>next_node</i> | the new next_node element |

Returns

calculatorNodeList* the address of the new data element

4.2 calculator.h File Reference

```
#include "share.h"
```

Data Structures

- struct [calculatorNodeList](#)
Data element of a [calculatorList](#), a number or an operator.
- struct [calculatorList](#)
[calculatorList](#) of tree nodes

Typedefs

- typedef struct [calculatorNodeList](#) [calculatorNodeList](#)

Functions

- [calculatorNodeList * newCalculatorNodeList](#) (bool is_number, double number, char operator, [calculatorNodeList](#) *previous_node, [calculatorNodeList](#) *next_node)
- void [deleteCalculatorNodeList](#) ([calculatorNodeList](#) *node)
- void [calculatorListInit](#) ([calculatorList](#) *list)
- void [calculatorListDelete](#) ([calculatorList](#) *list)
- bool [calculatorListEmpty](#) ([calculatorList](#) *list)
- bool [calculatorListOne](#) ([calculatorList](#) *list)
- bool [calculatorListLirst](#) ([calculatorList](#) *list)
- bool [calculatorListLast](#) ([calculatorList](#) *list)
- bool [calculatorListOutOfList](#) ([calculatorList](#) *list)
- void [calculatorListSetOnFirst](#) ([calculatorList](#) *list)
- void [calculatorListSetOnLast](#) ([calculatorList](#) *list)
- void [calculatorListNext](#) ([calculatorList](#) *list)

- void `calculatorListPrevious` (`calculatorList *list`)
- double `calculatorListGetCurrentNumber` (`calculatorList *list`)
- char `calculatorListGetCurrentOperator` (`calculatorList *list`)
- bool `calculatorListIsCurrentNumber` (`calculatorList *list`)
- void `calculatorListPrint` (`calculatorList *list`)
- bool `calculatorListInsertAfterCurrent` (`calculatorList *list`, bool `is_number`, double `number`, char `operator`)
- bool `calculatorListInsertBeforeCurrent` (`calculatorList *list`, bool `is_number`, double `number`, char `operator`)
- bool `calculatorListInsertLast` (`calculatorList *list`, bool `is_number`, double `number`, char `operator`)
- bool `calculatorListInsertLastFromString` (char `*string`, `calculatorList *list`)
- bool `calculatorListDeleteCurrent` (`calculatorList *list`)
- bool `calculatorListCalculate` (`calculatorList *list`, char `operator1`, char `operator2`)
- int `calculatorSearchNextOperatorString` (char `*string`)
- double `calculateFromString` (char `*string`)

4.2.1 Detailed Description

Author

Remi BERTHO

Date

18/02/15

Version

4.2.0

4.2.2 Typedef Documentation

4.2.2.1 typedef struct calculatorNodeList calculatorNodeList

4.2.3 Function Documentation

4.2.3.1 double calculateFromString (char * *string*)

Calculate the result of an expression passed in parameter

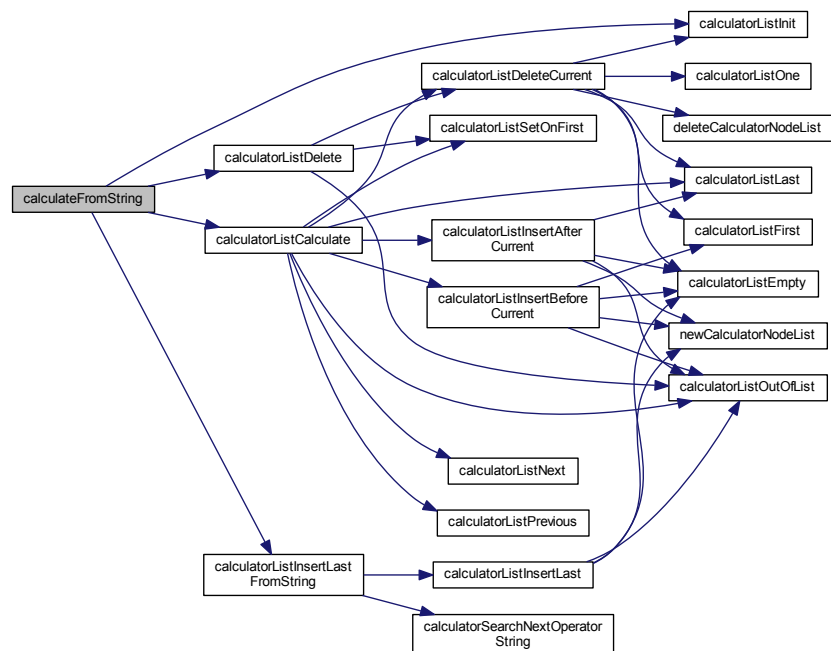
Parameters

| | | |
|----|---------------|--|
| in | <i>string</i> | the string which contain all the element of the list (a mathematical expression) |
|----|---------------|--|

Returns

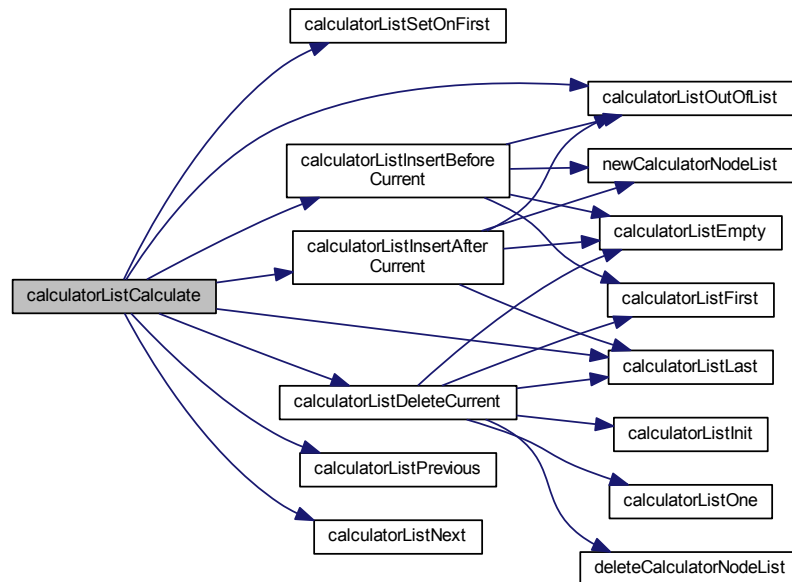
the result, Nan if the expression is incorrect

Here is the call graph for this function:



4.2.3.2 bool calculatorListCalculate (calculatorList * list, char operator1, char operator2)

Here is the call graph for this function:



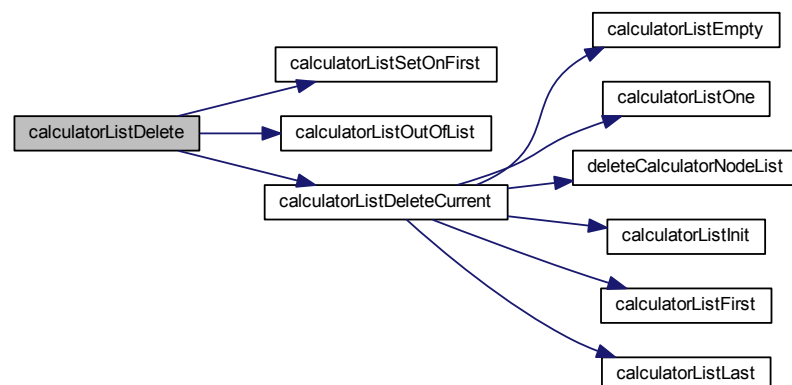
4.2.3.3 void calculatorListDelete (calculatorList * list)

Delete all the nodes of the list

Parameters

| | | |
|----|------|----------|
| in | list | the list |
|----|------|----------|

Here is the call graph for this function:



4.2.3.4 bool calculatorListDeleteCurrent (calculatorList * *list*)

Delete the current node to the list a set the current to the next or to the previous if the current was the last

Parameters

| | | |
|-----------|-------------|----------|
| <i>in</i> | <i>list</i> | the list |
|-----------|-------------|----------|

Returns

true if everything is OK, false otherwise

Calculate an expression with one of the two operator Scann all the list and if an operator is detected calculate the result and replace the expression to the result

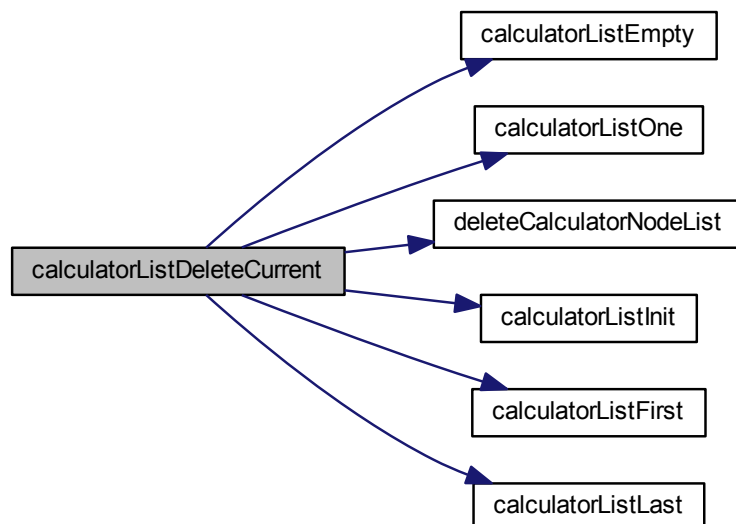
Parameters

| | | |
|-----------|-------------|----------|
| <i>in</i> | <i>list</i> | the list |
|-----------|-------------|----------|

Returns

true if everything is OK, false if there is no expression to evaluate

Here is the call graph for this function:

**4.2.3.5 bool calculatorListEmpty (calculatorList * list)**

Tests if it is an empty list.

Parameters

| | | |
|-----------|-------------|-----------------------|
| <i>in</i> | <i>list</i> | the list to be tested |
|-----------|-------------|-----------------------|

Returns

true if empty, false otherwise

4.2.3.6 double calculatorListGetCurrentNumber (calculatorList * *list*)

Return the number value stored in the current element.

Parameters

| | | |
|----|-------------|----------|
| in | <i>list</i> | the list |
|----|-------------|----------|

Returns

the number of the current element

4.2.3.7 char calculatorListGetCurrentOperator (calculatorList * *list*)

Return the operator value stored in the current element.

Parameters

| | | |
|----|-------------|----------|
| in | <i>list</i> | the list |
|----|-------------|----------|

Returns

the operator of the current element

4.2.3.8 void calculatorListInit (calculatorList * *list*)

Initialize the list structure members to be consistent with an empty list.

Parameters

| | | |
|-----|-------------|----------------------------|
| out | <i>list</i> | the list to be initialized |
|-----|-------------|----------------------------|

4.2.3.9 bool calculatorListInsertAfterCurrent (calculatorList * *list*, bool *is_number*, double *number*, char *operator*)

Insert a node after the current one

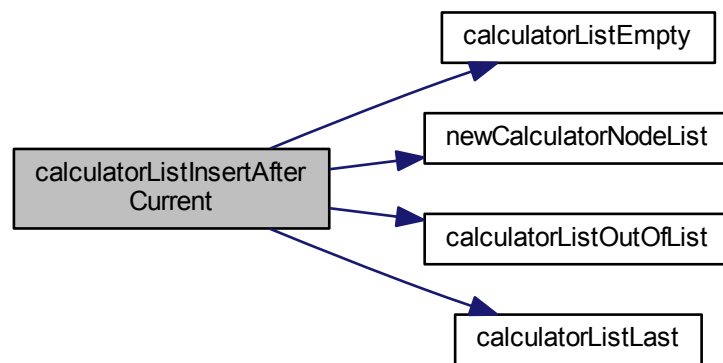
Parameters

| | | |
|----|------------------|----------------------------------|
| in | <i>is_number</i> | the new <i>is_number</i> element |
| in | <i>number</i> | the new number element |
| in | <i>operator</i> | the new operator element |
| in | <i>list</i> | the list |

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:



4.2.3.10 `bool calculatorListInsertBeforeCurrent (calculatorList * list, bool is_number, double number, char operator)`

Insert a node before the current one

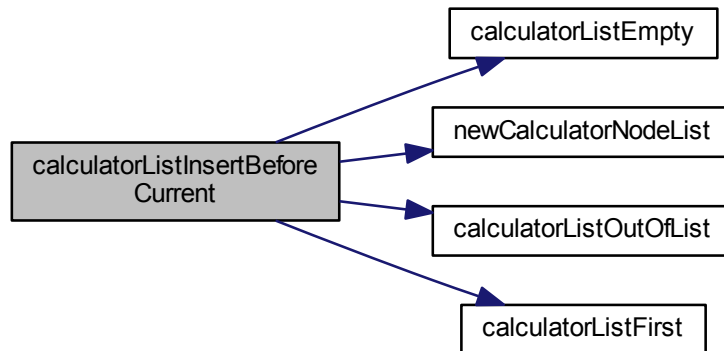
Parameters

| | | |
|----|------------------|---------------------------|
| in | <i>is_number</i> | the new is_number element |
| in | <i>number</i> | the new number element |
| in | <i>operator</i> | the new operator element |
| in | <i>list</i> | the list |

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:



4.2.3.11 `bool calculatorListInsertLast (calculatorList * list, bool is_number, double number, char operator)`

Insert a node in the last position of the list

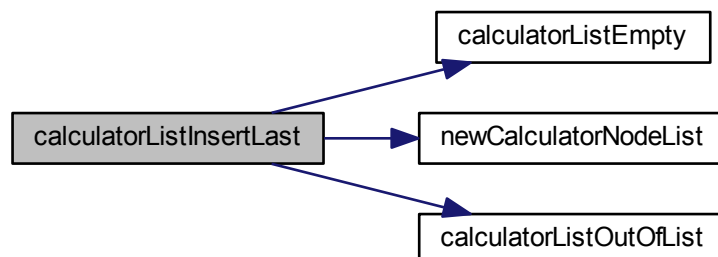
Parameters

| | | |
|----|------------------|---------------------------|
| in | <i>is_number</i> | the new is_number element |
| in | <i>number</i> | the new number element |
| in | <i>operator</i> | the new operator element |
| in | <i>list</i> | the list |

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:



4.2.3.12 bool calculatorListInsertLastFromString (char * *string*, calculatorList * *list*)

Convert the string into the list

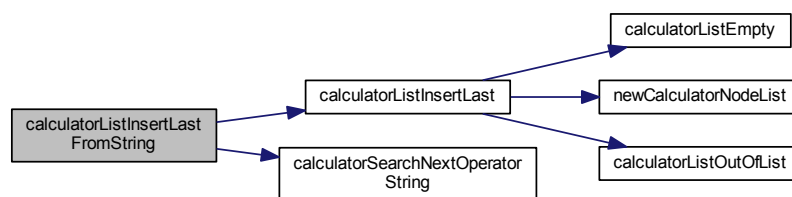
Parameters

| | | |
|----|---------------|--|
| in | <i>list</i> | the list |
| in | <i>string</i> | the string which contain all the element of the list (a mathematical expression) |

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:



4.2.3.13 bool calculatorListIsCurrentNumber (calculatorList * *list*)

Return the `ir_number` value stored in the current element.

Parameters

| | | |
|----|-------------|----------|
| in | <i>list</i> | the list |
|----|-------------|----------|

Returns

the `is_number` of the current element

4.2.3.14 bool calculatorListLast (calculatorList * *list*)

Tests if the current node is the last one.

Parameters

| | | |
|----|-------------|-----------------------|
| in | <i>list</i> | the list to be tested |
|----|-------------|-----------------------|

Returns

true if the current node is the last one, false otherwise

4.2.3.15 bool calculatorListFirst (calculatorList * *list*)

4.2.3.16 void calculatorListNext (calculatorList * *list*)

Set the current node on the next one.

Parameters

| | | |
|-----------|-------------|-------------------------|
| <i>in</i> | <i>list</i> | the list to be modified |
|-----------|-------------|-------------------------|

4.2.3.17 bool calculatorListOne (calculatorList * *list*)

Tests if there is only one element in the list

Parameters

| | | |
|-----------|-------------|-----------------------|
| <i>in</i> | <i>list</i> | the list to be tested |
|-----------|-------------|-----------------------|

Returns

true if there is only one element in the list, false otherwise

4.2.3.18 bool calculatorListOutOfList (calculatorList * *list*)

Tests if the current node is not valid (ie NULL).

Parameters

| | | |
|-----------|-------------|-----------------------|
| <i>in</i> | <i>list</i> | the list to be tested |
|-----------|-------------|-----------------------|

Returns

true if the current node is not valid, false otherwise

4.2.3.19 void calculatorListPrevious (calculatorList * *list*)

Set the current node on the previous one.

Parameters

| | | |
|-----------|-------------|-------------------------|
| <i>in</i> | <i>list</i> | the list to be modified |
|-----------|-------------|-------------------------|

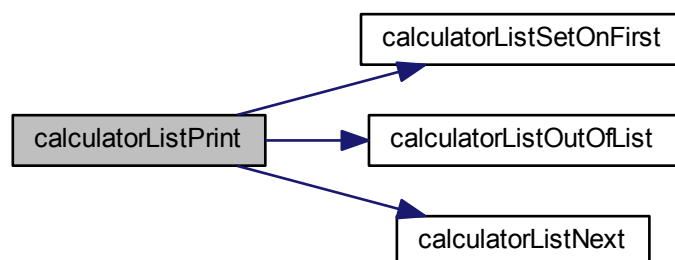
4.2.3.20 void calculatorListPrint (calculatorList * *list*)

Print the content of the list.

Parameters

| | | |
|-----------|-------------|--------------------------|
| <i>in</i> | <i>list</i> | the list to be displayed |
|-----------|-------------|--------------------------|

Here is the call graph for this function:



4.2.3.21 void calculatorListSetOnFirst (calculatorList * list)

Set the current node on the first one.

Parameters

| | | |
|-----------|-------------|-------------------------|
| <i>in</i> | <i>list</i> | the list to be modified |
|-----------|-------------|-------------------------|

4.2.3.22 void calculatorListSetOnLast (calculatorList * list)

Set the current node on the last one.

Parameters

| | | |
|-----------|-------------|-------------------------|
| <i>in</i> | <i>list</i> | the list to be modified |
|-----------|-------------|-------------------------|

4.2.3.23 int calculatorSearchNextOperatorString (char * string)

Search the next occurrence of an operator in a string

Parameters

| | | |
|-----------|---------------|--|
| <i>in</i> | <i>string</i> | the string which contain all the element of the list (a mathematical expression) |
|-----------|---------------|--|

Returns

the position of the next operator, 0 if there is no

4.2.3.24 void deleteCalculatorNodeList (calculatorNodeList * node)

Free a node

Parameters

| | | |
|-----|-------------|----------------------|
| out | <i>node</i> | the node to be freed |
|-----|-------------|----------------------|

4.2.3.25 `calculatorNodeList* newCalculatorNodeList (bool is_number, double number, char operator, calculatorNodeList * previous_node, calculatorNodeList * next_node)`

Allocate memory for the data element and initialize the structure members.

Parameters

| | | |
|----|----------------------|--------------------------------------|
| in | <i>is_number</i> | the new <i>is_number</i> element |
| in | <i>number</i> | the new <i>number</i> element |
| in | <i>operator</i> | the new <i>operator</i> element |
| in | <i>previous_node</i> | the new <i>previous_node</i> element |
| in | <i>next_node</i> | the new <i>next_node</i> element |

Returns

calculatorNodeList* the address of the new data element

4.3 csu_files.c File Reference

Files management.

```
#include "csu_files.h"
```

Functions

- FILE * [openFileCsuExtension](#) (char file_name[], char mode[])
- [csuStruct](#) * [readCsuFile](#) (char *file_name)
- bool [writeCsuFile](#) (char *file_name, [csuStruct](#) *ptr_csu_struct)
- bool [writeFileNewTurn](#) (char *file_name, [csuStruct](#) *ptr_csu_struct)
- bool [writeCsuXmlFile](#) (char *filename, [csuStruct](#) *ptr_csu_struct)
- void [addXmlFloatNode](#) (xmlNodePtr parent, char *name, float value, int decimal_place)
- void [addXmlBoolNode](#) (xmlNodePtr parent, char *name, int value)
- void [addXmlIntNode](#) (xmlNodePtr parent, char *name, int value)
- void [addXmlStringNode](#) (xmlNodePtr parent, char *name, char *value)
- void [addXmlFloatNodeIntProp](#) (xmlNodePtr parent, char *name, float value, int decimal_place, char *prop_name, int prop_value)
- [csuStruct](#) * [readCsuXmlFile](#) (char *filename)

4.3.1 Detailed Description

Files management.

Author

Remi BERTHO

Date

31/08/14

Version

4.2.0

4.3.2 Function Documentation

4.3.2.1 void addXmlBoolNode (xmlNodePtr *parent*, char * *name*, int *value*)

Add a XML node to the parent with a Boolean value

Parameters

| | | |
|----|---------------|---------------------------|
| in | <i>parent</i> | the parent node |
| in | <i>name</i> | the name of the new node |
| in | <i>value</i> | the value of the new node |

Here is the call graph for this function:

4.3.2.2 void addXmlFloatNode (xmlNodePtr *parent*, char * *name*, float *value*, int *decimal_place*)

Add a XML node to the parent with a float value and a number of decimal places

Parameters

| | | |
|----|----------------------|-----------------------------|
| in | <i>parent</i> | the parent node |
| in | <i>name</i> | the name of the new node |
| in | <i>value</i> | the value of the new node |
| in | <i>decimal_place</i> | the number of decimal place |

Here is the call graph for this function:

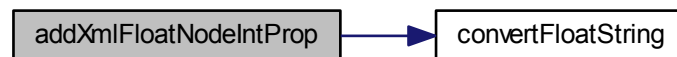
4.3.2.3 void addXmlFloatNodeIntProp (xmlNodePtr *parent*, char * *name*, float *value*, int *decimal_place*, char * *prop_name*, int *prop_value*)

Add a XML node to the parent with a float value and a number of decimal places and a integer property

Parameters

| | | |
|----|----------------------|-----------------------------|
| in | <i>parent</i> | the parent node |
| in | <i>name</i> | the name of the new node |
| in | <i>value</i> | the value of the new node |
| in | <i>decimal_place</i> | the number of decimal place |
| in | <i>prop_name</i> | the property's name |
| in | <i>prop_value</i> | the property's value |

Here is the call graph for this function:



4.3.2.4 void addXmlIntNode (xmlNodePtr *parent*, char * *name*, int *value*)

Add a XML node to the parent with a integer value

Parameters

| | | |
|----|---------------|---------------------------|
| in | <i>parent</i> | the parent node |
| in | <i>name</i> | the name of the new node |
| in | <i>value</i> | the value of the new node |

4.3.2.5 void addXmlStringNode (xmlNodePtr *parent*, char * *name*, char * *value*)

Add a XML node to the parent with a string value

Parameters

| | | |
|----|---------------|---------------------------|
| in | <i>parent</i> | the parent node |
| in | <i>name</i> | the name of the new node |
| in | <i>value</i> | the value of the new node |

4.3.2.6 FILE * openFileCsuExtension (char *file_name*[], char *mode*[])

Open a file with his name and with a specific mode and add the file extension if necessary.

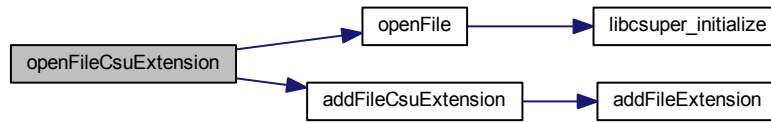
Parameters

| | | |
|----|---------------------|--------------|
| in | <i>file_name</i> [] | the filename |
| in | <i>mode</i> [] | the mode |

Returns

a pointer on the open file, NULL if there is a problem

Here is the call graph for this function:

**4.3.2.7 csuStruct * readCsuFile (char * file_name)**

Read the file with the name file_name and copy the result in a new csu structure.

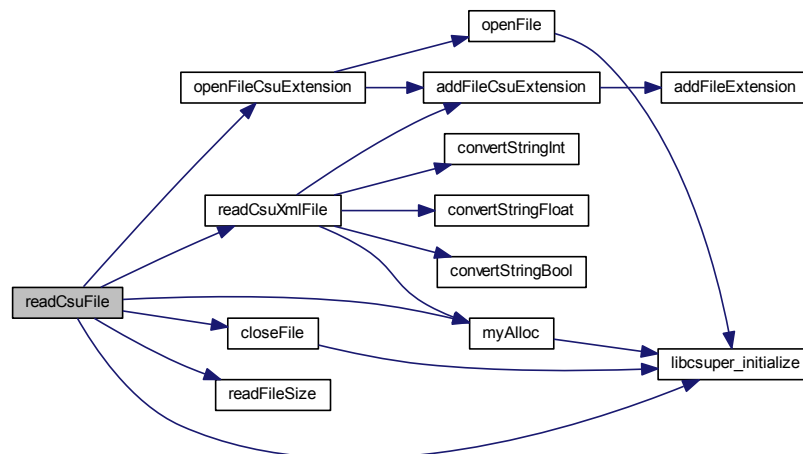
Parameters

| | | |
|----|-------------|--------------|
| in | file_name[] | the filename |
|----|-------------|--------------|

Returns

a pointer on the new csu structure, NULL if there is a problem

Here is the call graph for this function:

**4.3.2.8 csuStruct * readCsuXmlFile (char * filename)**

Read the xml file with the name file_name and copy the result in a new csu structure.

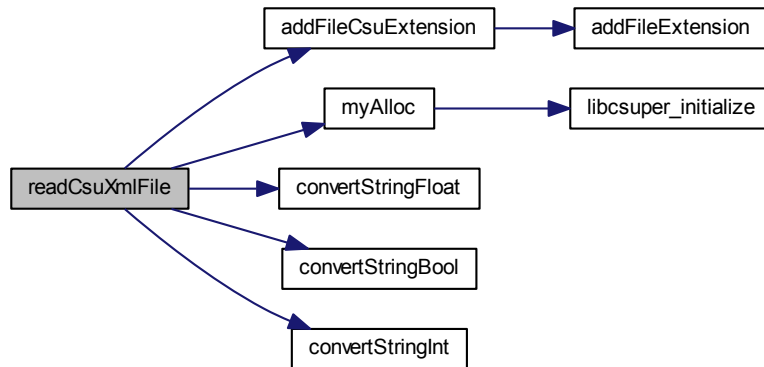
Parameters

| | | |
|----|-----------------|--------------|
| in | <i>filename</i> | the filename |
|----|-----------------|--------------|

Returns

a pointer on the new csu structure, NULL if there is a problem

Here is the call graph for this function:



4.3.2.9 bool writeCsuFile (char * file_name, csuStruct * ptr_csu_struct)

Write a csu file

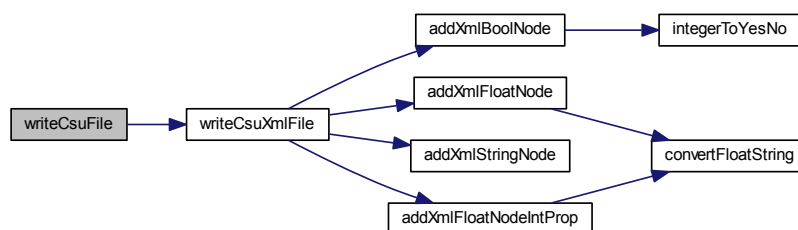
Parameters

| | | |
|----|------------------------|--|
| in | <i>*file_name</i> | the filename |
| in | <i>*ptr_csu_struct</i> | a pointer on a csuStruct |

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:



4.3.2.10 `bool writeCsuXmlFile (char * filename, csuStruct * ptr_csu_struct)`

Write a csu file in the new XML format

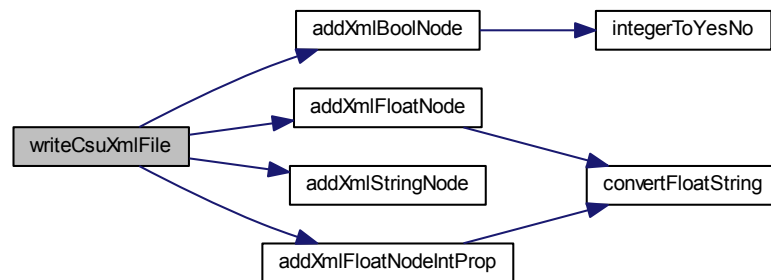
Parameters

| | | |
|----|-----------------------|--|
| in | <i>filename</i> | the filename |
| in | <i>ptr_csu_struct</i> | a pointer on a csuStruct |

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:



4.3.2.11 bool writeFileNewTurn (char * file_name, csuStruct * ptr_csu_struct)

Update the file with the new scores

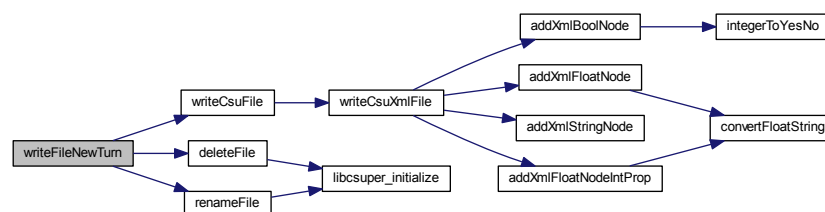
Parameters

| | | |
|----|------------------------|--|
| in | <i>*file_name</i> | the filename |
| in | <i>*ptr_csu_struct</i> | a pointer on a csuStruct |

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:



4.4 csu_files.h File Reference

Files management.

```
#include "csu_struct.h"
#include "filename.h"
#include <libxml/tree.h>
#include <unistd.h>
```

Macros

- `#define SIZE_MAX_FILE_NAME 1024`
- `#define FILE_EXTENSION_CSU "csu"`
- `#define STRING_CHECK_CSU_FILE "CompteurScoreUniversel"`

Functions

- `FILE * openFileCsuExtension (char file_name[], char mode[])`
- `csuStruct * readCsuFile (char *file_name)`
- `bool writeCsuFile (char *file_name, csuStruct *ptr_csu_struct)`
- `bool writeFileNewTurn (char *file_name, csuStruct *ptr_csu_struct)`
- `bool writeCsuXmlFile (char *filename, csuStruct *ptr_csu_struct)`
- `void addXmlFloatNode (xmlNodePtr parent, char *name, float value, int decimal_place)`
- `void addXmlBoolNode (xmlNodePtr parent, char *name, int value)`
- `void addXmlIntNode (xmlNodePtr parent, char *name, int value)`
- `void addXmlStringNode (xmlNodePtr parent, char *name, char *value)`
- `void addXmlFloatNodeIntProp (xmlNodePtr parent, char *name, float value, int decimal_place, char *prop_name, int prop_value)`
- `csuStruct * readCsuXmlFile (char *filename)`

4.4.1 Detailed Description

Files management.

Author

Remi BERTHO

Date

16/04/14

Version

2.2.0

4.4.2 Macro Definition Documentation

4.4.2.1 `#define FILE_EXTENSION_CSU "csu"`

Define the file extension to "csu"

4.4.2.2 `#define SIZE_MAX_FILE_NAME 1024`

Define the size maximum of a filename to 1024

4.4.2.3 #define STRING_CHECK_CSU_FILE "CompteurScoreUniversel"

String for checking if the file is a csu file.

4.4.3 Function Documentation

4.4.3.1 void addXmlBoolNode (xmlNodePtr *parent*, char * *name*, int *value*)

Add a XML node to the parent with a Boolean value

Parameters

| | | |
|----|---------------|---------------------------|
| in | <i>parent</i> | the parent node |
| in | <i>name</i> | the name of the new node |
| in | <i>value</i> | the value of the new node |

Here is the call graph for this function:



4.4.3.2 void addXmlFloatNode (xmlNodePtr *parent*, char * *name*, float *value*, int *decimal_place*)

Add a XML node to the parent with a float value and a number of decimal places

Parameters

| | | |
|----|----------------------|-----------------------------|
| in | <i>parent</i> | the parent node |
| in | <i>name</i> | the name of the new node |
| in | <i>value</i> | the value of the new node |
| in | <i>decimal_place</i> | the number of decimal place |

Here is the call graph for this function:



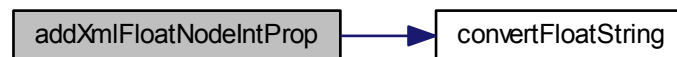
4.4.3.3 void addXmlFloatNodeIntProp (xmlNodePtr *parent*, char * *name*, float *value*, int *decimal_place*, char * *prop_name*, int *prop_value*)

Add a XML node to the parent with a float value and a number of decimal places and a integer property

Parameters

| | | |
|----|----------------------|-----------------------------|
| in | <i>parent</i> | the parent node |
| in | <i>name</i> | the name of the new node |
| in | <i>value</i> | the value of the new node |
| in | <i>decimal_place</i> | the number of decimal place |
| in | <i>prop_name</i> | the property's name |
| in | <i>prop_value</i> | the property's value |

Here is the call graph for this function:



4.4.3.4 void addXmlIntNode (xmlNodePtr *parent*, char * *name*, int *value*)

Add a XML node to the parent with a integer value

Parameters

| | | |
|----|---------------|---------------------------|
| in | <i>parent</i> | the parent node |
| in | <i>name</i> | the name of the new node |
| in | <i>value</i> | the value of the new node |

4.4.3.5 void addXmlStringNode (xmlNodePtr *parent*, char * *name*, char * *value*)

Add a XML node to the parent with a string value

Parameters

| | | |
|----|---------------|---------------------------|
| in | <i>parent</i> | the parent node |
| in | <i>name</i> | the name of the new node |
| in | <i>value</i> | the value of the new node |

4.4.3.6 FILE* openFileCsuExtension (char *file_name*[], char *mode*[])

Open a file with his name and with a specific mode and add the file extension if necessary.

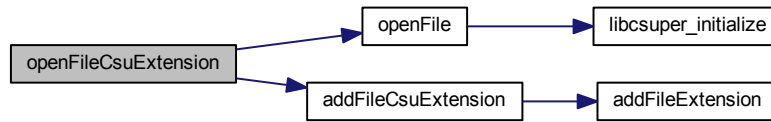
Parameters

| | | |
|----|---------------------|--------------|
| in | <i>file_name</i> [] | the filename |
| in | <i>mode</i> [] | the mode |

Returns

a pointer on the open file, NULL if there is a problem

Here is the call graph for this function:



4.4.3.7 csuStruct* readCsuFile (char * file_name)

Read the file with the name file_name and copy the result in a new csu structure.

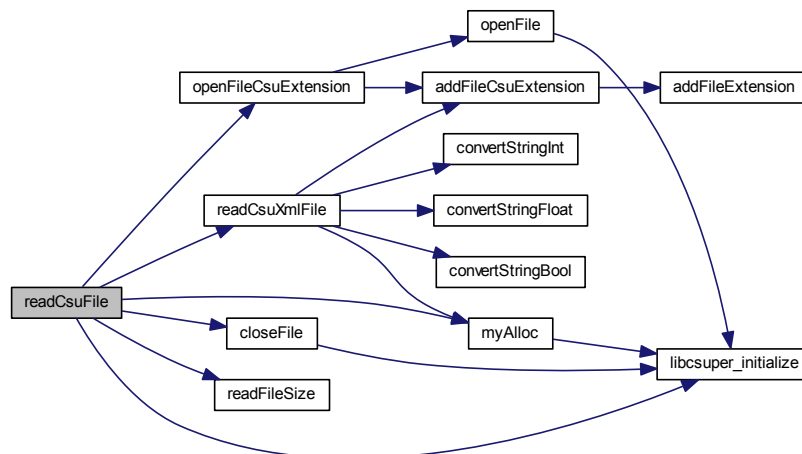
Parameters

| | | |
|----|-------------|--------------|
| in | file_name[] | the filename |
|----|-------------|--------------|

Returns

a pointer on the new csu structure, NULL if there is a problem

Here is the call graph for this function:



4.4.3.8 csuStruct* readCsuXmlFile (char * filename)

Read the xml file with the name file_name and copy the result in a new csu structure.

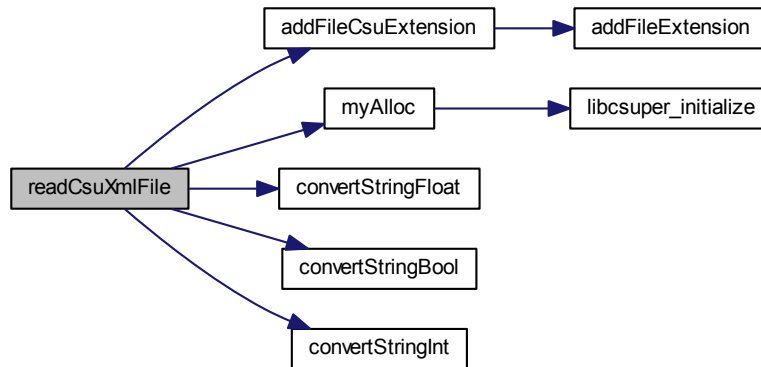
Parameters

| | | |
|----|-----------------|--------------|
| in | <i>filename</i> | the filename |
|----|-----------------|--------------|

Returns

a pointer on the new csu structure, NULL if there is a problem

Here is the call graph for this function:



4.4.3.9 bool writeCsuFile (char * file_name, csuStruct * ptr_csu_struct)

Write a csu file

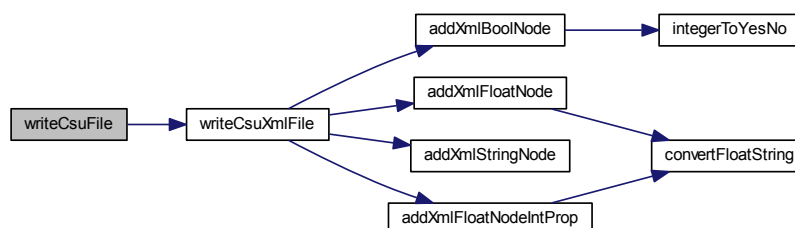
Parameters

| | | |
|----|------------------------|--|
| in | <i>*file_name</i> | the filename |
| in | <i>*ptr_csu_struct</i> | a pointer on a csuStruct |

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:



4.4.3.10 `bool writeCsuXmlFile (char * filename, csuStruct * ptr_csu_struct)`

Write a csu file in the new XML format

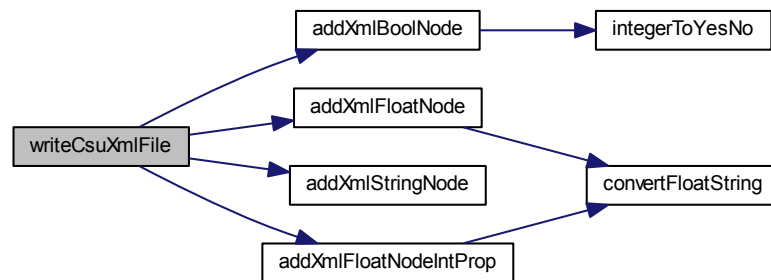
Parameters

| | | |
|----|-----------------------|--|
| in | <i>filename</i> | the filename |
| in | <i>ptr_csu_struct</i> | a pointer on a csuStruct |

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:



4.4.3.11 bool writeFileNewTurn (char * file_name, csuStruct * ptr_csu_struct)

Update the file with the new scores

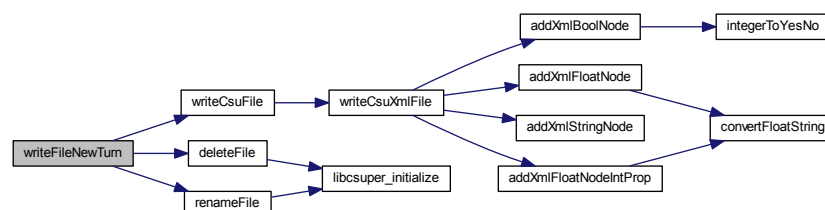
Parameters

| | | |
|----|------------------------|--|
| in | <i>*file_name</i> | the filename |
| in | <i>*ptr_csu_struct</i> | a pointer on a csuStruct |

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:



4.5 csu_struct.c File Reference

Management of the csu files.

```
#include "csu_struct.h"
```

Functions

- `csuStruct * newCsuStruct` (float nb_player, `game_config` config)
- void `closeCsuStruct` (`csuStruct *ptr_csu_struct`)
- void `startNewTurn` (`csuStruct *ptr_csu_struct`, int index_player)
- void `endNewTurn` (`csuStruct *ptr_csu_struct`, int index_player)
- void `rankCalculation` (`csuStruct *ptr_csu_struct`)
- int `searchIndexFromPosition` (`csuStruct *ptr_csu_struct`, int position, int *nb)
- void `addDistributorCsuStruct` (`csuStruct *ptr_csu_struct`, char *distributor_name)
- bool `exceedMaxNumber` (`csuStruct *ptr_csu_struct`)
- int `maxNbTurn` (`csuStruct *ptr_csu_struct`)
- int `searchPlayerIndex` (`csuStruct *ptr_csu_struct`, char *player_name)
- bool `differentsPlayerName` (`csuStruct *ptr_csu_struct`)
- `csuStruct * copyCsuStruct` (`csuStruct *ptr_csu_struct`)
- bool `changeDistributor` (`csuStruct *ptr_csu_struct`, int index)
- float `pointsAtTurn` (`csuStruct *ptr_csu_struct`, int player_index, int turn)
- int `rankAtTurn` (`csuStruct *ptr_csu_struct`, int player_index, int turn)
- int `lastRankAtTurn` (`csuStruct *ptr_csu_struct`, int turn)
- bool `deleteTurn` (`csuStruct *ptr_csu_struct`, int player_index, int turn)
- float `meanPoints` (`csuStruct *ptr_csu_struct`, int player_index)
- int `nbTurnBest` (`csuStruct *ptr_csu_struct`, int player_index)
- int `nbTurnWorst` (`csuStruct *ptr_csu_struct`, int player_index)
- int `nbTurnFirst` (`csuStruct *ptr_csu_struct`, int player_index)
- int `nbTurnLast` (`csuStruct *ptr_csu_struct`, int player_index)

4.5.1 Detailed Description

Management of the csu files.

Author

Remi BERTHO

Date

25/01/15

Version

4.1.0

4.5.2 Function Documentation

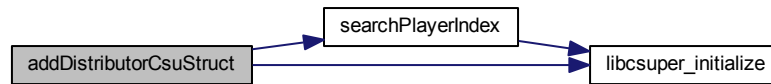
4.5.2.1 void addDistributorCsuStruct (`csuStruct * ptr_csu_struct`, char * `distributor_name`)

Add the distributor on the structure

Parameters

| | | |
|----|--------------------------------|--|
| in | <i>*distributor_↔ name</i> | the name of the distributor |
| in | <i>*ptr_csu_struct</i> | a pointer on a csuStruct |

Here is the call graph for this function:



4.5.2.2 bool changeDistributor (csuStruct * ptr_csu_struct, int index)

Change the distributor

Parameters

| | | |
|----|------------------------|--|
| in | <i>*ptr_csu_struct</i> | a pointer on a csuStruct |
| in | <i>index</i> | the index of the player |

Returns

true if the distributor can be change, false otherwise

4.5.2.3 void closeCsuStruct (csuStruct * ptr_csu_struct)

Free a [csuStruct](#)

Parameters

| | | |
|---------|------------------------|--|
| in, out | <i>*ptr_csu_struct</i> | a pointer to the csuStruct |
|---------|------------------------|--|

4.5.2.4 csuStruct * copyCsuStruct (csuStruct * ptr_csu_struct)

Copy a csu structure

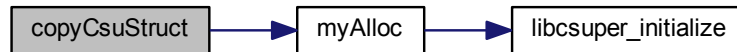
Parameters

| | | |
|----|------------------------|--|
| in | <i>*ptr_csu_struct</i> | a pointer on a csuStruct |
|----|------------------------|--|

Returns

a pointer on the new csu structure

Here is the call graph for this function:



4.5.2.5 bool deleteTurn (csuStruct * ptr_csu_struct, int player_index, int turn)

Delete a turn of a player or all of them

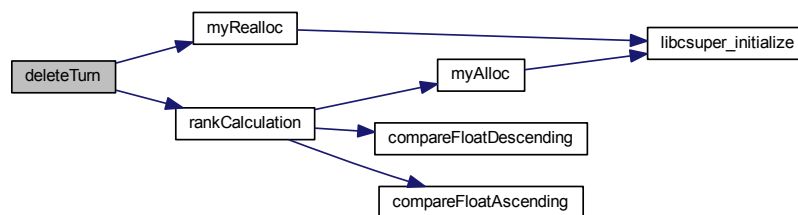
Parameters

| | | |
|----|------------------------|--|
| in | <i>*ptr_csu_struct</i> | a pointer on a csuStruct |
| in | <i>player_index</i> | the index of the player |
| in | <i>turn</i> | the turn |

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:



4.5.2.6 bool differentsPlayerName (csuStruct * ptr_csu_struct)

Search if all the name are different

Parameters

| | | |
|----|------------------------|--|
| in | <i>*ptr_csu_struct</i> | a pointer on a csuStruct |
|----|------------------------|--|

Returns

true if all player names are different, false otherwise

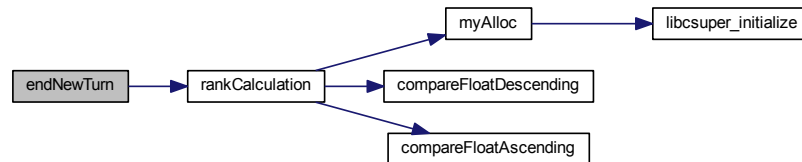
4.5.2.7 void endNewTurn (csuStruct * *ptr_csu_struct*, int *index_player*)

Update the total points, the number of turn, the distributor and the rank for a new turn

Parameters

| | | |
|---------|------------------------|---|
| in, out | <i>*ptr_csu_struct</i> | a pointer on a csuStruct |
| in, out | <i>index_player</i> | index_player the index of the player who begin a new turn, -1 if everybody begin a new turn |

Here is the call graph for this function:



4.5.2.8 bool exceedMaxNumber (csuStruct * ptr_csu_struct)

Check if someone exceed the maximum number

Parameters

| | | |
|----|------------------------|--|
| in | <i>*ptr_csu_struct</i> | a pointer on a csuStruct |
|----|------------------------|--|

Returns

true if someone exceed, false otherwise

4.5.2.9 int lastRankAtTurn (csuStruct * ptr_csu_struct, int turn)

Return the last rank at a specific turn

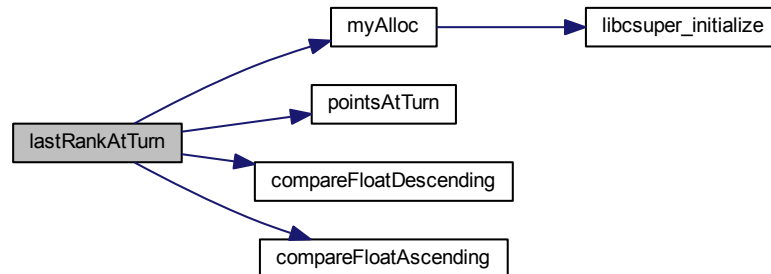
Parameters

| | | |
|----|------------------------|--|
| in | <i>*ptr_csu_struct</i> | a pointer on a csuStruct |
| in | <i>turn</i> | she turn |

Returns

the last rank or 0 if the game configuration is not turn based

Here is the call graph for this function:



4.5.2.10 int maxNbTurn (csuStruct * ptr_csu_struct)

Search the maximal number of turn

Parameters

| | | |
|----|------------------------|--|
| in | <i>*ptr_csu_struct</i> | a pointer on a csuStruct |
|----|------------------------|--|

Returns

the maximal number of turn

4.5.2.11 float meanPoints (csuStruct * ptr_csu_struct, int player_index)

Calculate the mean points of a player

Parameters

| | | |
|----|------------------------|--|
| in | <i>*ptr_csu_struct</i> | a pointer on a csuStruct |
| in | <i>player_index</i> | the index of the player |

Returns

the mean points

4.5.2.12 int nbTurnBest (csuStruct * ptr_csu_struct, int player_index)

Calculate the number of turn where the player do the best score

Parameters

| | | |
|----|-----------------------|--|
| in | <i>*ptr_cs_struct</i> | a pointer on a csuStruct |
| in | <i>player_index</i> | the index of the player |

Returns

the number of turn where the player do the best score

4.5.2.13 int nbTurnFirst ([csuStruct](#) * *ptr_cs_struct*, int *player_index*)

Calculate the number of turn where the player is the first place

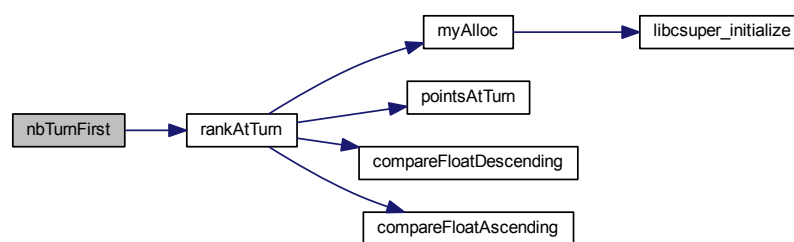
Parameters

| | | |
|----|-----------------------|--|
| in | <i>*ptr_cs_struct</i> | a pointer on a csuStruct |
| in | <i>player_index</i> | the index of the player |

Returns

the number of turn where the player is the first

Here is the call graph for this function:



4.5.2.14 int nbTurnLast ([csuStruct](#) * *ptr_cs_struct*, int *player_index*)

Calculate the number of turn where the player is the last place

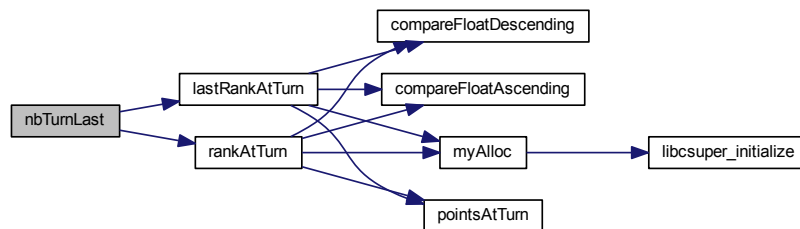
Parameters

| | | |
|----|-----------------------|--|
| in | <i>*ptr_cs_struct</i> | a pointer on a csuStruct |
| in | <i>player_index</i> | the index of the player |

Returns

the number of turn where the player is the last

Here is the call graph for this function:



4.5.2.15 int nbTurnWorst (csuStruct * ptr_csu_struct, int player_index)

Calculate the number of turn where the player do the worst score

Parameters

| | | |
|----|------------------------|--|
| in | <i>*ptr_csu_struct</i> | a pointer on a csuStruct |
| in | <i>player_index</i> | the index of the player |

Returns

the number of turn where the player do the worst score

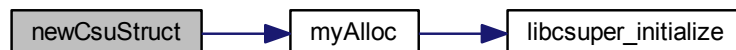
4.5.2.16 csuStruct * newCsuStruct (float nb_player, game_config config)

Create a new [csuStruct](#) from a game configuration and the number of player.

Parameters

| | | |
|----|------------------|------------------------|
| in | <i>nb_player</i> | the number of player |
| in | <i>config</i> | the game configuration |

Here is the call graph for this function:



4.5.2.17 float pointsAtTurn (csuStruct * ptr_csu_struct, int player_index, int turn)

Return the number of points of a player at a specific turn

Parameters

| | | |
|----|------------------------|--|
| in | <i>*ptr_csu_struct</i> | a pointer on a csuStruct |
| in | <i>player_index</i> | the index of the player |
| in | <i>turn</i> | the turn |

Returns

the total number of points

4.5.2.18 float rankAtTurn ([csuStruct](#) * *ptr_csu_struct*, int *player_index*, int *turn*)

Return the ranking of a player at a specific turn

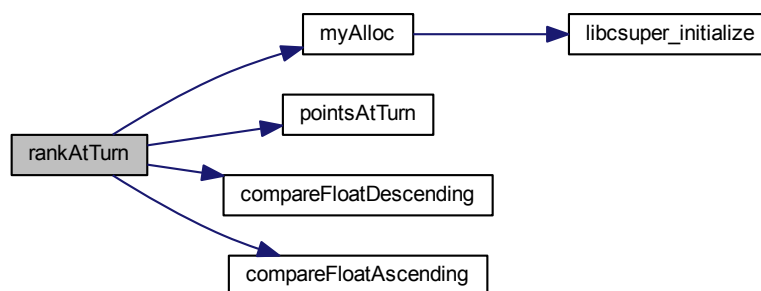
Parameters

| | | |
|----|------------------------|--|
| in | <i>*ptr_csu_struct</i> | a pointer on a csuStruct |
| in | <i>player_index</i> | the index of the player |
| in | <i>turn</i> | she turn |

Returns

the ranking or 0 if the game configuration is not turn based

Here is the call graph for this function:



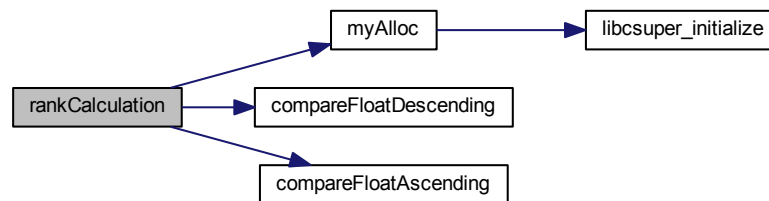
4.5.2.19 void rankCalculation ([csuStruct](#) * *ptr_csu_struct*)

Calculate the rank

Parameters

| | | |
|---------|------------------------|--|
| in, out | <i>*ptr_csu_struct</i> | a pointer on a csuStruct |
|---------|------------------------|--|

Here is the call graph for this function:



4.5.2.20 int searchIndexFromPosition (csuStruct * ptr_csu_struct, int position, int * nb)

Search the index in the array of the person who is the 'position' position

Parameters

| | | |
|---------|------------------------|--|
| in, out | <i>*ptr_csu_struct</i> | a pointer on a csuStruct |
| in, out | <i>position</i> | the position |
| in, out | <i>nb</i> | the nbth player who have the position will be selected |

Returns

the index or NULL if the position doesn't exist

Here is the call graph for this function:



4.5.2.21 int searchPlayerIndex (csuStruct * ptr_csu_struct, char * player_name)

Search the index of a person

Parameters

| | | |
|----|------------------------|--|
| in | <i>*player_name</i> | the name of the player |
| in | <i>*ptr_csu_struct</i> | a pointer on a csuStruct |

Returns

the index, -1 if there is not found

Here is the call graph for this function:



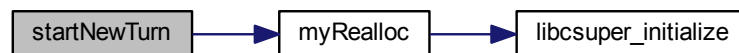
4.5.2.22 void startNewTurn (csuStruct * ptr_csu_struct, int index_player)

Reallocate the memory for the point to begin a new turn.

Parameters

| | | |
|---------|-----------------|--|
| in, out | *ptr_csu_struct | a pointer on a csuStruct |
| in, out | index_player | the index of the player who begin a new turn, -1 if everybody begin a new turn |

Here is the call graph for this function:



4.6 csu_struct.h File Reference

Management of the csu files header.

```

#include <time.h>
#include <float.h>
#include "share.h"
#include "file.h"
  
```

Data Structures

- struct [game_config](#)
- struct [csuStruct](#)

Macros

- #define [SIZE_MAX_NAME](#) 30
- #define [VERSION](#) 1.5

Functions

- `csuStruct * newCsuStruct` (float nb_player, `game_config` config)
- void `closeCsuStruct` (`csuStruct *ptr_csu_struct`)
- void `startNewTurn` (`csuStruct *ptr_csu_struct`, int index_player)
- void `endNewTurn` (`csuStruct *ptr_csu_struct`, int index_player)
- void `rankCalculation` (`csuStruct *ptr_csu_struct`)
- int `searchIndexFromPosition` (`csuStruct *ptr_csu_struct`, int position, int *nb)
- void `addDistributorCsuStruct` (`csuStruct *ptr_csu_struct`, char *distributor_name)
- bool `exceedMaxNumber` (`csuStruct *ptr_csu_struct`)
- int `maxNbTurn` (`csuStruct *ptr_csu_struct`)
- int `searchPlayerIndex` (`csuStruct *ptr_csu_struct`, char *player_name)
- bool `differentsPlayerName` (`csuStruct *ptr_csu_struct`)
- `csuStruct * copyCsuStruct` (`csuStruct *ptr_csu_struct`)
- bool `changeDistributor` (`csuStruct *ptr_csu_struct`, int index)
- float `pointsAtTurn` (`csuStruct *ptr_csu_struct`, int player_index, int turn)
- int `rankAtTurn` (`csuStruct *ptr_csu_struct`, int player_index, int turn)
- int `lastRankAtTurn` (`csuStruct *ptr_csu_struct`, int turn)
- bool `deleteTurn` (`csuStruct *ptr_csu_struct`, int player_index, int turn)
- float `meanPoints` (`csuStruct *ptr_csu_struct`, int player_index)
- int `nbTurnBest` (`csuStruct *ptr_csu_struct`, int player_index)
- int `nbTurnWorst` (`csuStruct *ptr_csu_struct`, int player_index)
- int `nbTurnFirst` (`csuStruct *ptr_csu_struct`, int player_index)
- int `nbTurnLast` (`csuStruct *ptr_csu_struct`, int player_index)

4.6.1 Detailed Description

Management of the csu files header.

Author

Remi BERTHO

Date

25/01/15

Version

4.01.0

4.6.2 Macro Definition Documentation

4.6.2.1 #define SIZE_MAX_NAME 30

Define size max of name

4.6.2.2 #define VERSION 1.5

Define the file's version

4.6.3 Function Documentation

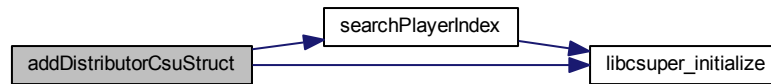
4.6.3.1 void addDistributorCsuStruct (csuStruct * ptr_csu_struct, char * distributor_name)

Add the distributor on the structure

Parameters

| | | |
|----|--------------------------------|--|
| in | <i>*distributor_↔ name</i> | the name of the distributor |
| in | <i>*ptr_csu_struct</i> | a pointer on a csuStruct |

Here is the call graph for this function:



4.6.3.2 bool changeDistributor (csuStruct * ptr_csu_struct, int index)

Change the distributor

Parameters

| | | |
|----|------------------------|--|
| in | <i>*ptr_csu_struct</i> | a pointer on a csuStruct |
| in | <i>index</i> | the index of the player |

Returns

true if the distributor can be change, false otherwise

4.6.3.3 void closeCsuStruct (csuStruct * ptr_csu_struct)

Free a [csuStruct](#)

Parameters

| | | |
|---------|------------------------|--|
| in, out | <i>*ptr_csu_struct</i> | a pointer to the csuStruct |
|---------|------------------------|--|

4.6.3.4 csuStruct* copyCsuStruct (csuStruct * ptr_csu_struct)

Copy a csu structure

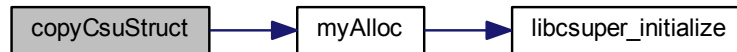
Parameters

| | | |
|----|------------------------|--|
| in | <i>*ptr_csu_struct</i> | a pointer on a csuStruct |
|----|------------------------|--|

Returns

a pointer on the new csu structure

Here is the call graph for this function:



4.6.3.5 bool deleteTurn (csuStruct * ptr_csu_struct, int player_index, int turn)

Delete a turn of a player or all of them

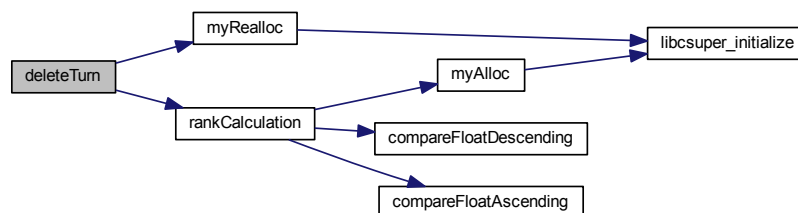
Parameters

| | | |
|----|------------------------|--|
| in | <i>*ptr_csu_struct</i> | a pointer on a csuStruct |
| in | <i>player_index</i> | the index of the player |
| in | <i>turn</i> | the turn |

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:



4.6.3.6 bool differentsPlayerName (csuStruct * ptr_csu_struct)

Search if all the name are different

Parameters

| | | |
|----|------------------------|--|
| in | <i>*ptr_csu_struct</i> | a pointer on a csuStruct |
|----|------------------------|--|

Returns

true if all player names are different, false otherwise

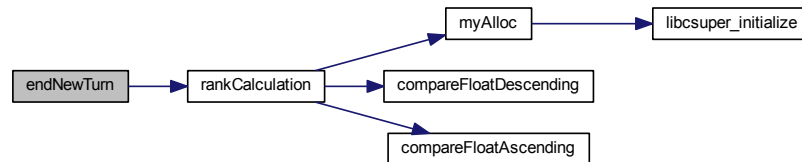
4.6.3.7 void endNewTurn (**csuStruct** * *ptr_csu_struct*, int *index_player*)

Update the total points, the number of turn, the distributor and the rank for a new turn

Parameters

| | | |
|---------|------------------------|---|
| in, out | <i>*ptr_csu_struct</i> | a pointer on a csuStruct |
| in, out | <i>index_player</i> | index_player the index of the player who begin a new turn, -1 if everybody begin a new turn |

Here is the call graph for this function:



4.6.3.8 bool exceedMaxNumber (csuStruct * ptr_csu_struct)

Check if someone exceed the maximum number

Parameters

| | | |
|----|------------------------|--|
| in | <i>*ptr_csu_struct</i> | a pointer on a csuStruct |
|----|------------------------|--|

Returns

true if someone exceed, false otherwise

4.6.3.9 int lastRankAtTurn (csuStruct * ptr_csu_struct, int turn)

Return the last rank at a specific turn

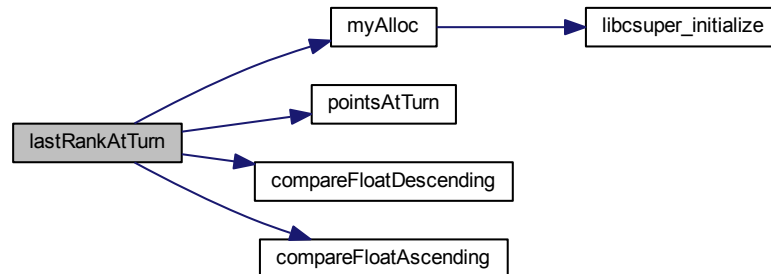
Parameters

| | | |
|----|------------------------|--|
| in | <i>*ptr_csu_struct</i> | a pointer on a csuStruct |
| in | <i>turn</i> | she turn |

Returns

the last rank or 0 if the game configuration is not turn based

Here is the call graph for this function:



4.6.3.10 int maxNbTurn (csuStruct * ptr_csu_struct)

Search the maximal number of turn

Parameters

| | | |
|----|-----------------|--|
| in | *ptr_csu_struct | a pointer on a csuStruct |
|----|-----------------|--|

Returns

the maximal number of turn

4.6.3.11 float meanPoints (csuStruct * ptr_csu_struct, int player_index)

Calculate the mean points of a player

Parameters

| | | |
|----|-----------------|--|
| in | *ptr_csu_struct | a pointer on a csuStruct |
| in | player_index | the index of the player |

Returns

the mean points

4.6.3.12 int nbTurnBest (csuStruct * ptr_csu_struct, int player_index)

Calculate the number of turn where the player do the best score

Parameters

| | | |
|----|-----------------|--|
| in | *ptr_csu_struct | a pointer on a csuStruct |
| in | player_index | the index of the player |

| | | |
|----|------------------------|--|
| in | <i>*ptr_csu_struct</i> | a pointer on a csuStruct |
| in | <i>player_index</i> | the index of the player |

Returns

the number of turn where the player do the best score

4.6.3.13 int nbTurnFirst (csuStruct * ptr_csu_struct, int player_index)

Calculate the number of turn where the player is the first place

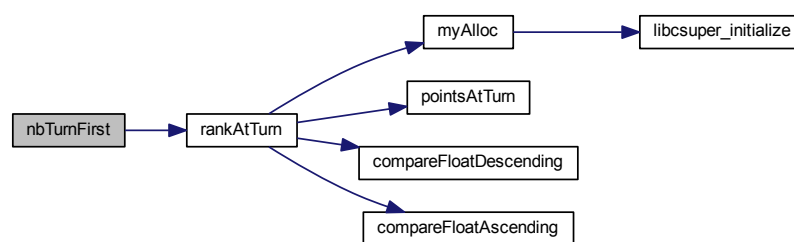
Parameters

| | | |
|----|------------------------|--|
| in | <i>*ptr_csu_struct</i> | a pointer on a csuStruct |
| in | <i>player_index</i> | the index of the player |

Returns

the number of turn where the player is the first

Here is the call graph for this function:



4.6.3.14 int nbTurnLast (csuStruct * ptr_csu_struct, int player_index)

Calculate the number of turn where the player is the last place

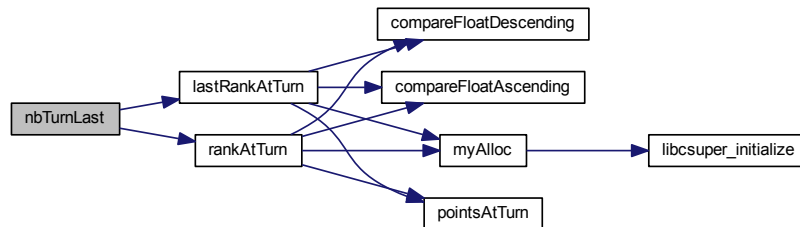
Parameters

| | | |
|----|------------------------|--|
| in | <i>*ptr_csu_struct</i> | a pointer on a csuStruct |
| in | <i>player_index</i> | the index of the player |

Returns

the number of turn where the player is the last

Here is the call graph for this function:



4.6.3.15 int nbTurnWorst (csuStruct * ptr_csu_struct, int player_index)

Calculate the number of turn where the player do the worst score

Parameters

| | | |
|----|------------------------|--|
| in | <i>*ptr_csu_struct</i> | a pointer on a csuStruct |
| in | <i>player_index</i> | the index of the player |

Returns

the number of turn where the player do the worst score

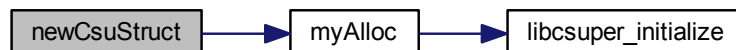
4.6.3.16 csuStruct* newCsuStruct (float nb_player, game_config config)

Create a new [csuStruct](#) from a game configuration and the number of player.

Parameters

| | | |
|----|------------------|------------------------|
| in | <i>nb_player</i> | the number of player |
| in | <i>config</i> | the game configuration |

Here is the call graph for this function:



4.6.3.17 float pointsAtTurn (csuStruct * ptr_csu_struct, int player_index, int turn)

Return the number of points of a player at a specific turn

Parameters

| | | |
|----|------------------------|--|
| in | <i>*ptr_csu_struct</i> | a pointer on a csuStruct |
| in | <i>player_index</i> | the index of the player |
| in | <i>turn</i> | the turn |

Returns

the total number of points

4.6.3.18 int rankAtTurn (csuStruct * ptr_csu_struct, int player_index, int turn)

Return the ranking of a player at a specific turn

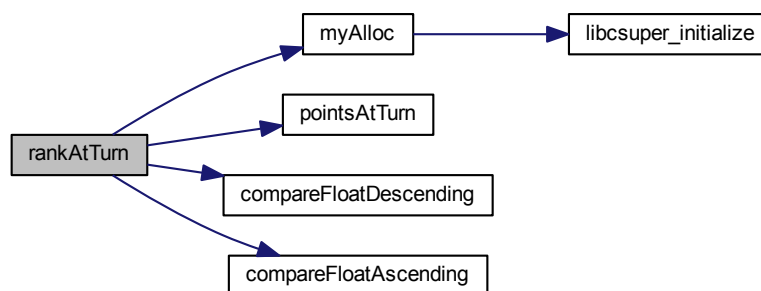
Parameters

| | | |
|----|------------------------|--|
| in | <i>*ptr_csu_struct</i> | a pointer on a csuStruct |
| in | <i>player_index</i> | the index of the player |
| in | <i>turn</i> | she turn |

Returns

the ranking or 0 if the game configuration is not turn based

Here is the call graph for this function:

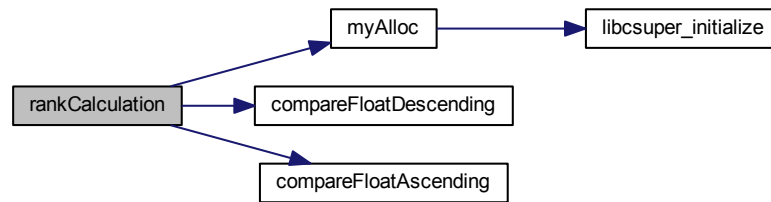
**4.6.3.19** void rankCalculation (csuStruct * ptr_csu_struct)

Calculate the rank

Parameters

| | | |
|---------|------------------------|--|
| in, out | <i>*ptr_csu_struct</i> | a pointer on a csuStruct |
|---------|------------------------|--|

Here is the call graph for this function:



4.6.3.20 int searchIndexFromPosition (csuStruct * ptr_csu_struct, int position, int * nb)

Search the index in the array of the person who is the 'position' position

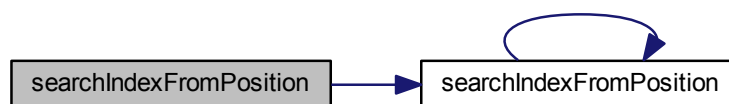
Parameters

| | | |
|---------|------------------------|--|
| in, out | <i>*ptr_csu_struct</i> | a pointer on a csuStruct |
| in, out | <i>position</i> | the position |
| in, out | <i>nb</i> | the nbth player who have the position will be selected |

Returns

the index or NULL if the position doesn't exist

Here is the call graph for this function:



4.6.3.21 int searchPlayerIndex (csuStruct * ptr_csu_struct, char * player_name)

Search the index of a person

Parameters

| | | |
|----|------------------------|--|
| in | <i>*player_name</i> | the name of the player |
| in | <i>*ptr_csu_struct</i> | a pointer on a csuStruct |

Returns

the index, -1 if there is not found

Here is the call graph for this function:

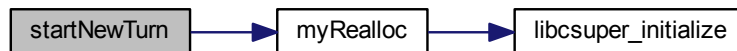
**4.6.3.22 void startNewTurn (csuStruct * ptr_csu_struct, int index_player)**

Reallocate the memory for the point to begin a new turn.

Parameters

| | | |
|---------|------------------------|--|
| in, out | <i>*ptr_csu_struct</i> | a pointer on a csuStruct |
| in, out | <i>index_player</i> | the index of the player who begin a new turn, -1 if everybody begin a new turn |

Here is the call graph for this function:

**4.7 export.c File Reference**

Export function.

```
#include "export.h"
```

Functions

- void [errorHandler](#) (HPDF_STATUS error_no, HPDF_STATUS detail_no, void *user_data)
- bool [canUseUtf8Pdf](#) (void)
- void [pdfShowText](#) (HPDF_Page page, char *text, [export_pdf](#) *ptr_export_pdf)
- bool [pdfTextOutTable](#) (HPDF_Page page, float pos_min_x, float pos_y, const char *text, float max_width, int ranking, [export_pdf](#) *ptr_export_pdf)
- bool [exportToPdf](#) (csuStruct *ptr_csu_struct, char *filename)
- bool [initializePdfExport](#) ([export_pdf](#) *ptr_export_pdf, csuStruct *ptr_csu_struct)
- void [closeExportPdf](#) ([export_pdf](#) *ptr_export_pdf)
- void [printPointsPdf](#) (HPDF_Page page, float *pos_y, csuStruct *ptr_csu_struct, [export_pdf](#) *ptr_export_pdf, float table_width)

- void [printNamesPdf](#) ([export_pdf](#) *ptr_export_pdf, [csuStruct](#) *ptr_csu_struct, float *pos_y, float table_width, HPDF_Page page)
- void [printLegendPdf](#) ([export_pdf](#) *ptr_export_pdf, [csuStruct](#) *ptr_csu_struct, float *pos_y, float table_width, HPDF_Page page)
- bool [createFirstPagePdf](#) ([export_pdf](#) *ptr_export_pdf, [csuStruct](#) *ptr_csu_struct, char *filename)
- float [tableWidthCalculatePdf](#) ([export_pdf](#) *ptr_export_pdf, [csuStruct](#) *ptr_csu_struct, HPDF_Page page)
- void [createPdfGrid](#) (HPDF_Page page, float top_x, float top_y, float bottom_x, float bottom_y, float length_row, float length_column)
- void [addTotalPointsRankingPdf](#) (HPDF_Page page, [csuStruct](#) *ptr_csu_struct, float y, [export_pdf](#) *ptr_export_pdf)
- void [addStatsPdf](#) (HPDF_Page page, [csuStruct](#) *ptr_csu_struct, float y, [export_pdf](#) *ptr_export_pdf)
- bool [createOtherPagePdf](#) ([export_pdf](#) *ptr_export_pdf, [csuStruct](#) *ptr_csu_struct)
- bool [addPodiumPdf](#) (HPDF_Page page, [csuStruct](#) *ptr_csu_struct, float y, [export_pdf](#) *ptr_export_pdf)
- bool [exportToCsv](#) ([csuStruct](#) *ptr_csu_struct, char *filename)
- bool [exportToM](#) ([csuStruct](#) *ptr_csu_struct, char *filename)

4.7.1 Detailed Description

Export function.

Author

Remi BERTHO

Date

22/12/14

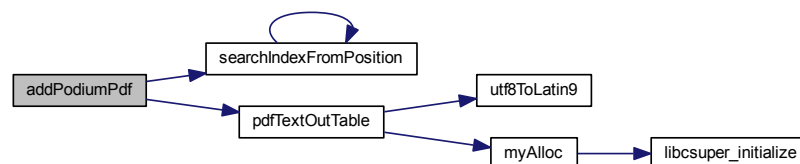
Version

4.1.0

4.7.2 Function Documentation

4.7.2.1 bool [addPodiumPdf](#) (HPDF_Page *page*, [csuStruct](#) * *ptr_csu_struct*, float *y*, [export_pdf](#) * *ptr_export_pdf*)

Here is the call graph for this function:



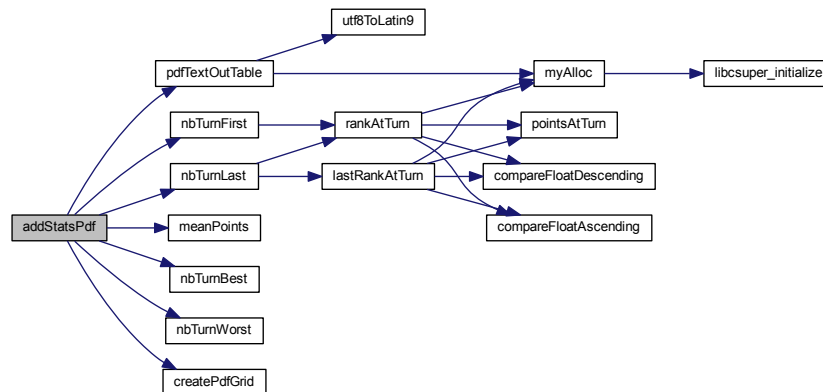
4.7.2.2 void [addStatsPdf](#) (HPDF_Page *page*, [csuStruct](#) * *ptr_csu_struct*, float *y*, [export_pdf](#) * *ptr_export_pdf*)

Print the stats on a pdf page

Parameters

| | | |
|----|-----------------------|---|
| in | <i>ptr_csu_struct</i> | a pointer on a csuStruct |
| in | <i>page</i> | the pdf page |
| in | <i>y</i> | the top y coordinate |
| in | <i>ptr_export_pdf</i> | a pointer on a export_pdf |

Here is the call graph for this function:



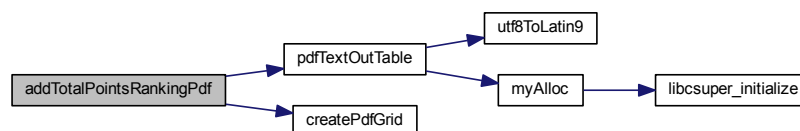
4.7.2.3 void addTotalPointsRankingPdf (HPDF_Page page, csuStruct * ptr_csu_struct, float y, export_pdf * ptr_export_pdf)

Print the total points and the ranking on a pdf page

Parameters

| | | |
|----|-----------------------|---|
| in | <i>ptr_csu_struct</i> | a pointer on a csuStruct |
| in | <i>page</i> | the pdf page |
| in | <i>y</i> | the top y coordinate |
| in | <i>ptr_export_pdf</i> | a pointer on a export_pdf |

Here is the call graph for this function:



4.7.2.4 bool canUseUtf8Pdf (void)

Say if you can use UTF-8 in a pdf file or not

Returns

true if you can use UF-8, false otherwise

4.7.2.5 void closeExportPdf (export_pdf * ptr_export_pdf)

Close the structure [export_pdf](#)

Parameters

| | | |
|----|-----------------------|---|
| in | <i>ptr_export_pdf</i> | a pointer on a export_pdf |
|----|-----------------------|---|

4.7.2.6 bool createFirstPagePdf (export_pdf * ptr_export_pdf, csuStruct * ptr_csu_struct, char * filename)

Create the first page of the pdf

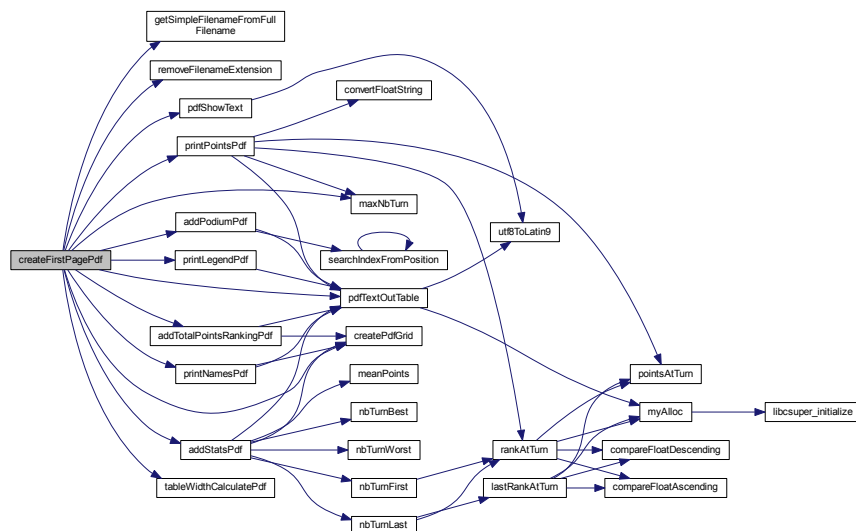
Parameters

| | | |
|----|-----------------------|---|
| in | <i>ptr_csu_struct</i> | a pointer on a csuStruct |
| in | <i>ptr_export_pdf</i> | a pointer on a export_pdf |
| in | <i>filename</i> | the filename of the pdf file |

Returns

true if it need another page, false otherwise

Here is the call graph for this function:



4.7.2.7 bool createOtherPagePdf (export_pdf * ptr_export_pdf, csuStruct * ptr_csu_struct)

Create the other page of the pdf

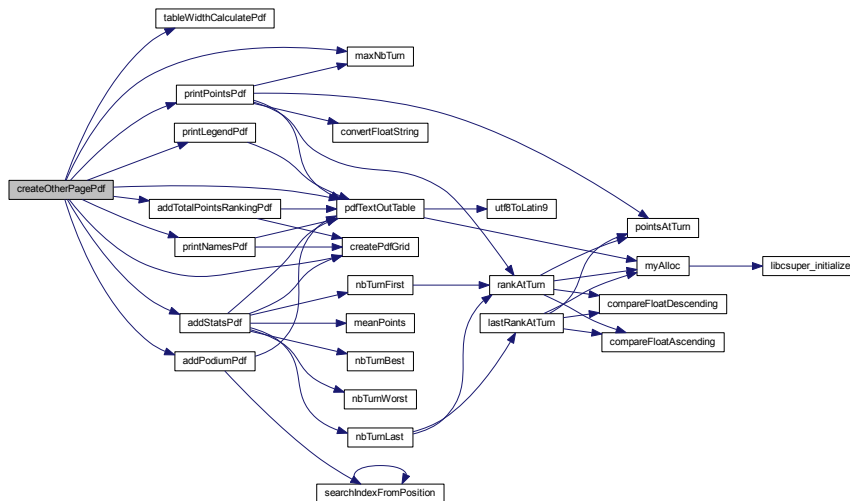
Parameters

| | | |
|----|-----------------------|---|
| in | <i>ptr_csu_struct</i> | a pointer on a csuStruct |
| in | <i>ptr_export_pdf</i> | a pointer on a export_pdf |

Returns

true if it need another page, false otherwise

Here is the call graph for this function:



4.7.2.8 void createPdfGrid (HPDF_Page *page*, float *top_x*, float *top_y*, float *bottom_x*, float *bottom_y*, float *length_row*, float *length_column*)

Print a grid to a pdf page

Parameters

| | | |
|----|----------------------|---|
| in | <i>page</i> | the pdf page |
| in | <i>top_x</i> | the x coordinate of the top left corn |
| in | <i>top_y</i> | the y coordinate of the top left corn |
| in | <i>bottom_x</i> | the x coordinate of the bottom right corn |
| in | <i>bottom_y</i> | the y coordinate of the bottom right corn |
| in | <i>length_row</i> | the length of the row |
| in | <i>length_column</i> | the length of the column |

4.7.2.9 void errorHandler (HPDF_STATUS *error_no*, HPDF_STATUS *detail_no*, void * *user_data*)

Print an error message when an error occurred on libhpdf

Parameters

| | | |
|----|-----------------|------------------|
| in | <i>error_no</i> | the error number |
|----|-----------------|------------------|

| | | |
|----|------------------|-------------------|
| in | <i>detail_no</i> | the detail number |
| in | <i>user_data</i> | user data |

4.7.2.10 bool exportToCsv (csuStruct * ptr_csu_struct, char * filename)

Export the csu structure to a csv file

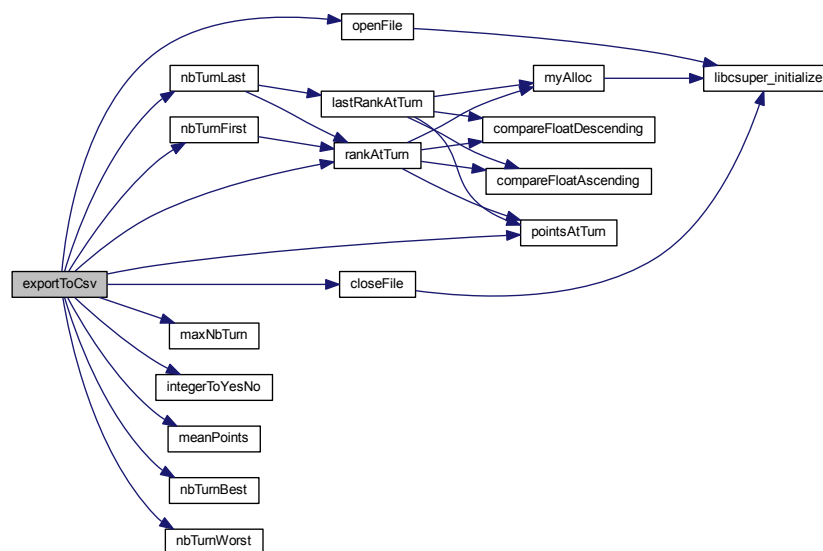
Parameters

| | | |
|----|-----------------------|--|
| in | <i>ptr_csu_struct</i> | a pointer on a csuStruct |
| in | <i>filename</i> | the filename |

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:



4.7.2.11 bool exportToM (csuStruct * ptr_csu_struct, char * filename)

Export the csu structure to a m file (octave/matlab file)

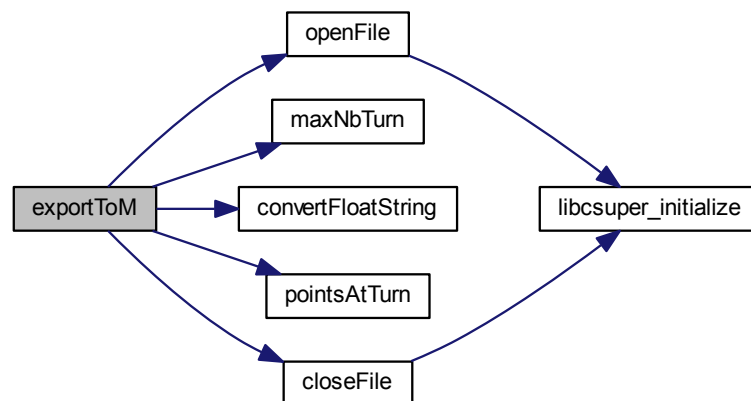
Parameters

| | | |
|----|-----------------------|--|
| in | <i>ptr_csu_struct</i> | a pointer on a csuStruct |
| in | <i>filename</i> | the filename |

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:

**4.7.2.12** `bool exportToPdf (csuStruct * ptr_csu_struct, char * filename)`

Export the csu structure to a pdf

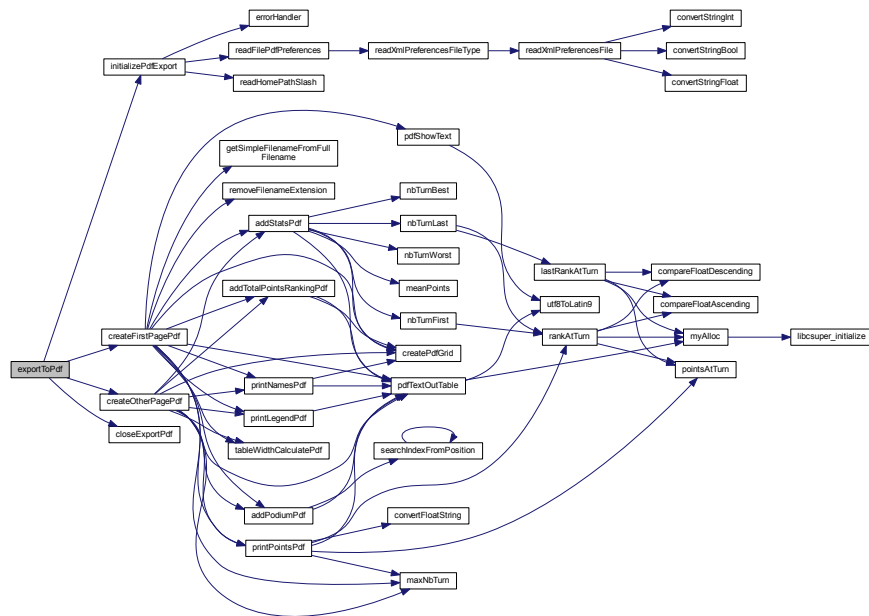
Parameters

| | | |
|----|-----------------------|--|
| in | <i>ptr_csu_struct</i> | a pointer on a csuStruct |
| in | <i>filename</i> | the filename of the pdf file |

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:



4.7.2.13 bool initializePdfExport (export_pdf * ptr_export_pdf, csuStruct * ptr_csu_struct)

Initialize the exportation

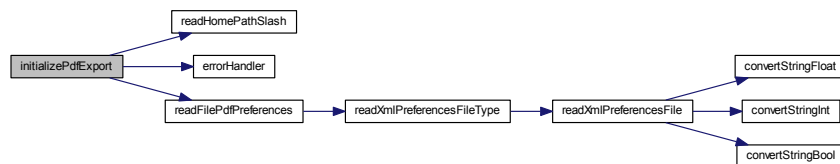
Parameters

| | | |
|----|----------------|---|
| in | ptr_csu_struct | a pointer on a csuStruct |
| in | ptr_export_pdf | a pointer on a export_pdf |

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:



4.7.2.14 void pdfShowText (HPDF_Page page, char * text, export_pdf * ptr_export_pdf)

Print the text in the page

Parameters

| | | |
|----|-----------------------|---|
| in | <i>page</i> | the page |
| in | <i>text</i> | the text to print |
| in | <i>ptr_export_pdf</i> | a pointer on a export_pdf |

Here is the call graph for this function:



4.7.2.15 `bool pdfTextOutTable (HPDF_Page page, float pos_min_x, float pos_y, const char * text, float max_width, int ranking, export_pdf * ptr_export_pdf)`

Print the text in the page center with a maximum width

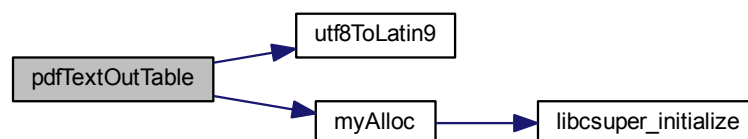
Parameters

| | | |
|----|-----------------------|---|
| in | <i>page</i> | the page |
| in | <i>pos_min_x</i> | the minimum position on the x axis |
| in | <i>pos_y</i> | the position on the y axis |
| in | <i>text</i> | the text to print |
| in | <i>max_width</i> | the maximum width of the text |
| in | <i>ranking</i> | the ranking of the player to determine the color of the text, put 0 to print in black |
| in | <i>ptr_export_pdf</i> | a pointer on a export_pdf |

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:



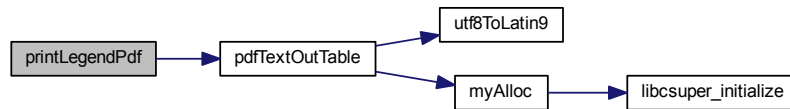
4.7.2.16 `void printLegendPdf (export_pdf * ptr_export_pdf, csuStruct * ptr_csu_struct, float * pos_y, float table_width, HPDF_Page page)`

Print the legend on a pdf page if needed

Parameters

| | | |
|----|-----------------------|---|
| in | <i>page</i> | the page |
| in | <i>pos_y</i> | a pointer to the first position on the y axis |
| in | <i>table_width</i> | the width of a table |
| in | <i>ptr_csu_struct</i> | a pointer on a csuStruct |
| in | <i>ptr_export_pdf</i> | a pointer on a export_pdf |

Here is the call graph for this function:



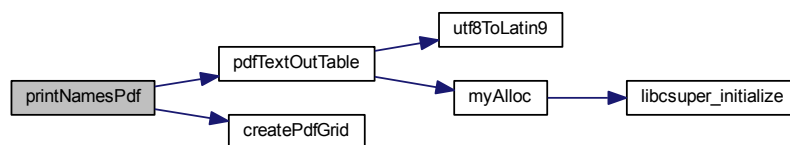
4.7.2.17 void printNamesPdf ([export_pdf](#) * *ptr_export_pdf*, [csuStruct](#) * *ptr_csu_struct*, float * *pos_y*, float *table_width*, [HPDF_Page](#) *page*)

Print the names on a pdf page

Parameters

| | | |
|----|-----------------------|---|
| in | <i>page</i> | the page |
| in | <i>pos_y</i> | a pointer to the first position on the y axis |
| in | <i>table_width</i> | the width of a table |
| in | <i>ptr_csu_struct</i> | a pointer on a csuStruct |
| in | <i>ptr_export_pdf</i> | a pointer on a export_pdf |

Here is the call graph for this function:



4.7.2.18 void printPointsPdf ([HPDF_Page](#) *page*, float * *pos_y*, [csuStruct](#) * *ptr_csu_struct*, [export_pdf](#) * *ptr_export_pdf*, float *table_width*)

Print the points on a pdf page

Parameters

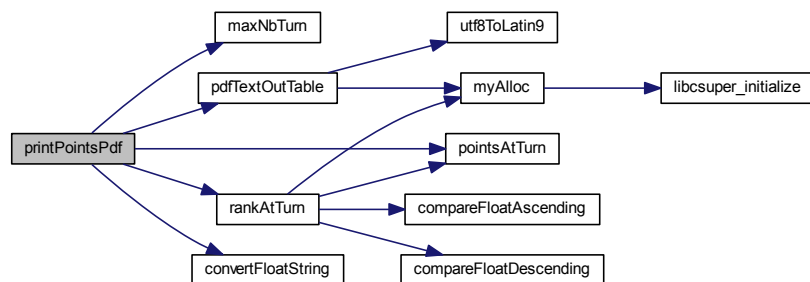
| | | |
|----|-------------|----------|
| in | <i>page</i> | the page |
|----|-------------|----------|

| | | |
|----|-----------------------|---|
| in | <i>pos_y</i> | a pointer to the first position on the y axis |
| in | <i>table_width</i> | the width of a table |
| in | <i>ptr_csu_struct</i> | a pointer on a csuStruct |
| in | <i>ptr_export_pdf</i> | a pointer on a export_pdf |

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:



4.7.2.19 float tableWidthCalculatePdf (export_pdf * ptr_export_pdf, csuStruct * ptr_csu_struct, HPDF_Page page)

Calculate the table width

Parameters

| | | |
|----|-----------------------|---|
| in | <i>page</i> | the page |
| in | <i>ptr_csu_struct</i> | a pointer on a csuStruct |
| in | <i>ptr_export_pdf</i> | a pointer on a export_pdf |

Returns

the table width

4.8 export.h File Reference

Header for the export function.

```

#include "share.h"
#include "preferences_files.h"
#include "csu_struct.h"
#include "csu_files.h"

```

Data Structures

- struct [export_pdf_preferences](#)
- struct [export_pdf](#)

Macros

- `#define` `DEFAULT_MARGIN` 40
- `#define` `DEFAULT_FONT_SIZE` 12
- `#define` `TEXT_BUFFER_SIZE` 1024

Enumerations

- `enum` `CharacterSetPdf` { `UTF8`, `ISO885915` }

Functions

- `void` `errorHandler` (`HPDF_STATUS` error_no, `HPDF_STATUS` detail_no, `void` *user_data)
- `bool` `canUseUtf8Pdf` (`void`)
- `void` `pdfShowText` (`HPDF_Page` page, `char` *text, `export_pdf` *ptr_export_pdf)
- `bool` `pdfTextOutTable` (`HPDF_Page` page, `float` pos_min_x, `float` pos_y, `const char` *text, `float` max_width, `int` ranking, `export_pdf` *ptr_export_pdf)
- `bool` `exportToPdf` (`csuStruct` *ptr_csu_struct, `char` *filename)
- `bool` `initializePdfExport` (`export_pdf` *ptr_export_pdf, `csuStruct` *ptr_csu_struct)
- `void` `closeExportPdf` (`export_pdf` *ptr_export_pdf)
- `void` `printPointsPdf` (`HPDF_Page` page, `float` *pos_y, `csuStruct` *ptr_csu_struct, `export_pdf` *ptr_export_pdf, `float` table_width)
- `void` `printNamesPdf` (`export_pdf` *ptr_export_pdf, `csuStruct` *ptr_csu_struct, `float` *pos_y, `float` table_width, `HPDF_Page` page)
- `void` `printLegendPdf` (`export_pdf` *ptr_export_pdf, `csuStruct` *ptr_csu_struct, `float` *pos_y, `float` table_width, `HPDF_Page` page)
- `bool` `createFirstPagePdf` (`export_pdf` *ptr_export_pdf, `csuStruct` *ptr_csu_struct, `char` *filename)
- `float` `tableWidthCalculatePdf` (`export_pdf` *ptr_export_pdf, `csuStruct` *ptr_csu_struct, `HPDF_Page` page)
- `void` `createPdfGrid` (`HPDF_Page` page, `float` top_x, `float` top_y, `float` bottom_x, `float` bottom_y, `float` length_row, `float` length_column)
- `void` `addTotalPointsRankingPdf` (`HPDF_Page` page, `csuStruct` *ptr_csu_struct, `float` y, `export_pdf` *ptr_export_pdf)
- `void` `addStatsPdf` (`HPDF_Page` page, `csuStruct` *ptr_csu_struct, `float` y, `export_pdf` *ptr_export_pdf)
- `bool` `createOtherPagePdf` (`export_pdf` *ptr_export_pdf, `csuStruct` *ptr_csu_struct)
- `bool` `addPodiumPdf` (`HPDF_Page` page, `csuStruct` *ptr_csu_struct, `float` y, `export_pdf` *ptr_export_pdf)
- `bool` `exportToCsv` (`csuStruct` *ptr_csu_struct, `char` *filename)
- `bool` `exportToM` (`csuStruct` *ptr_csu_struct, `char` *filename)

4.8.1 Detailed Description

Header for the export function.

Author

Remi BERTHO

Date

22/12/14

Version

4.1.0

4.8.2 Macro Definition Documentation

4.8.2.1 `#define DEFAULT_FONT_SIZE 12`

Define the default font size

4.8.2.2 `#define DEFAULT_MARGIN 40`

Define the default margin

4.8.2.3 `#define TEXT_BUFFER_SIZE 1024`

Define the size of the text buffer

4.8.3 Enumeration Type Documentation

4.8.3.1 `enum CharacterSetPdf`

Enumerator

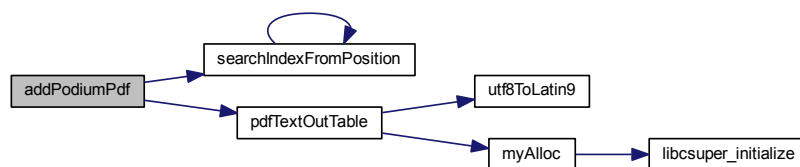
UTF8

ISO885915

4.8.4 Function Documentation

4.8.4.1 `bool addPodiumPdf (HPDF_Page page, csuStruct * ptr_csu_struct, float y, export_pdf * ptr_export_pdf)`

Here is the call graph for this function:



4.8.4.2 `void addStatsPdf (HPDF_Page page, csuStruct * ptr_csu_struct, float y, export_pdf * ptr_export_pdf)`

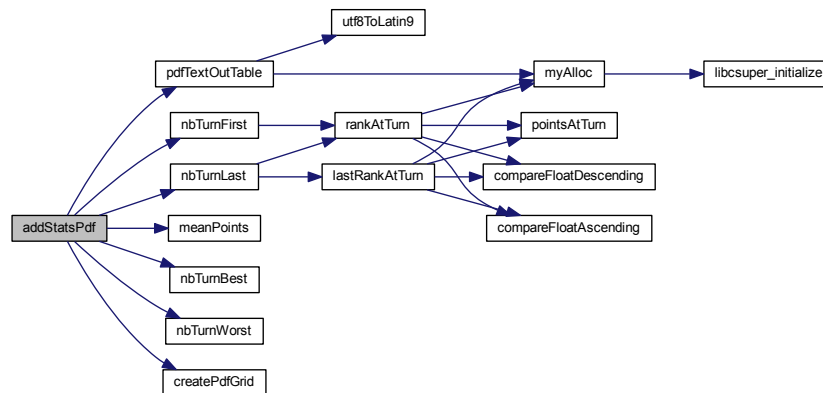
Print the stats on a pdf page

Parameters

| | | |
|----|-----------------------|--|
| in | <i>ptr_csu_struct</i> | a pointer on a csuStruct |
| in | <i>page</i> | the pdf page |
| in | <i>y</i> | the top y coordinate |

| | | |
|----|-----------------------|---|
| in | <i>ptr_export_pdf</i> | a pointer on a export_pdf |
|----|-----------------------|---|

Here is the call graph for this function:



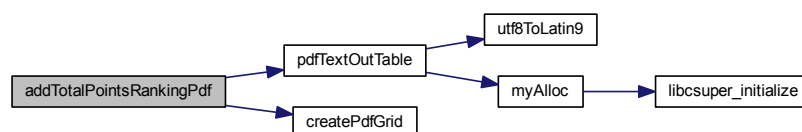
4.8.4.3 void addTotalPointsRankingPdf (HPDF_Page page, csuStruct * ptr_csu_struct, float y, export_pdf * ptr_export_pdf)

Print the total points and the ranking on a pdf page

Parameters

| | | |
|----|-----------------------|---|
| in | <i>ptr_csu_struct</i> | a pointer on a csuStruct |
| in | <i>page</i> | the pdf page |
| in | <i>y</i> | the top y coordinate |
| in | <i>ptr_export_pdf</i> | a pointer on a export_pdf |

Here is the call graph for this function:



4.8.4.4 bool canUseUtf8Pdf (void)

Say if you can use UTF-8 in a pdf file or not

Returns

true if you can use UF-8, false otherwise

4.8.4.5 void closeExportPdf (export_pdf * ptr_export_pdf)

Close the structure [export_pdf](#)

Parameters

| | | |
|----|-----------------------|---|
| in | <i>ptr_export_pdf</i> | a pointer on a export_pdf |
|----|-----------------------|---|

4.8.4.6 `bool createFirstPagePdf (export_pdf * ptr_export_pdf, csuStruct * ptr_csu_struct, char * filename)`

Create the first page of the pdf

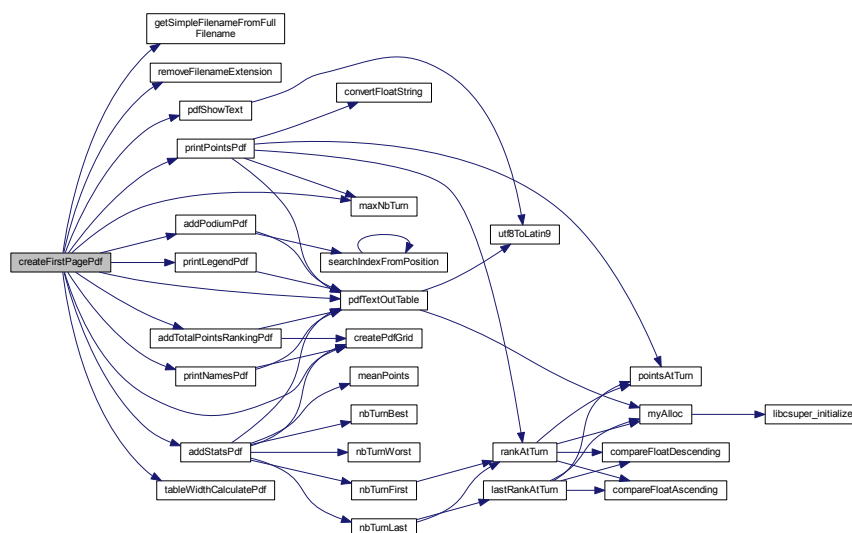
Parameters

| | | |
|----|-----------------------|---|
| in | <i>ptr_csu_struct</i> | a pointer on a csuStruct |
| in | <i>ptr_export_pdf</i> | a pointer on a export_pdf |
| in | <i>filename</i> | the filename of the pdf file |

Returns

true if it need another page, false otherwise

Here is the call graph for this function:

4.8.4.7 `bool createOtherPagePdf (export_pdf * ptr_export_pdf, csuStruct * ptr_csu_struct)`

Create the other page of the pdf

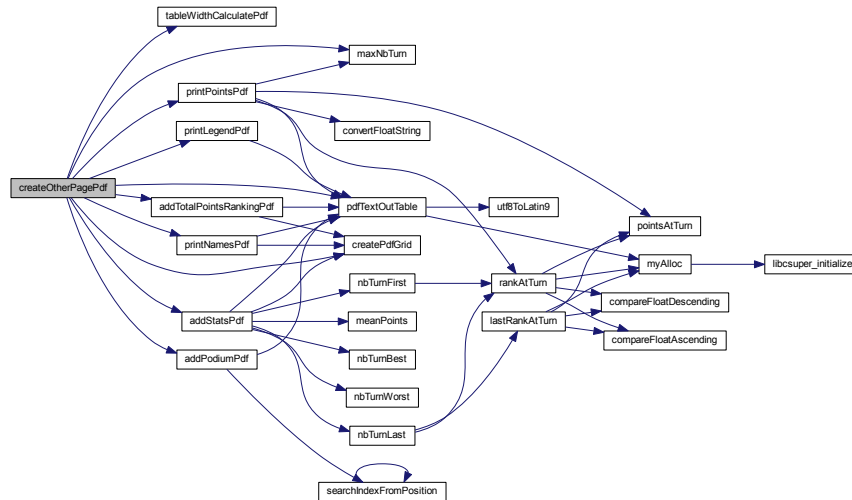
Parameters

| | | |
|----|-----------------------|---|
| in | <i>ptr_csu_struct</i> | a pointer on a csuStruct |
| in | <i>ptr_export_pdf</i> | a pointer on a export_pdf |

Returns

true if it need another page, false otherwise

Here is the call graph for this function:



4.8.4.8 void createPdfGrid (HPDF_Page page, float top_x, float top_y, float bottom_x, float bottom_y, float length_row, float length_column)

Print a grid to a pdf page

Parameters

| | | |
|----|---------------|---|
| in | page | the pdf page |
| in | top_x | the x coordinate of the top left corn |
| in | top_y | the y coordinate of the top left corn |
| in | bottom_x | the x coordinate of the bottom right corn |
| in | bottom_y | the y coordinate of the bottom right corn |
| in | length_row | the length of the row |
| in | length_column | the length of the column |

4.8.4.9 void errorHandler (HPDF_STATUS error_no, HPDF_STATUS detail_no, void * user_data)

Print an error message when an error occured on libhpdf

Parameters

| | | |
|----|-----------|-------------------|
| in | error_no | the error number |
| in | detail_no | the detail number |
| in | user_data | user data |

4.8.4.10 bool exportToCsv (csuStruct * ptr_csu_struct, char * filename)

Export the csu structure to a csv file

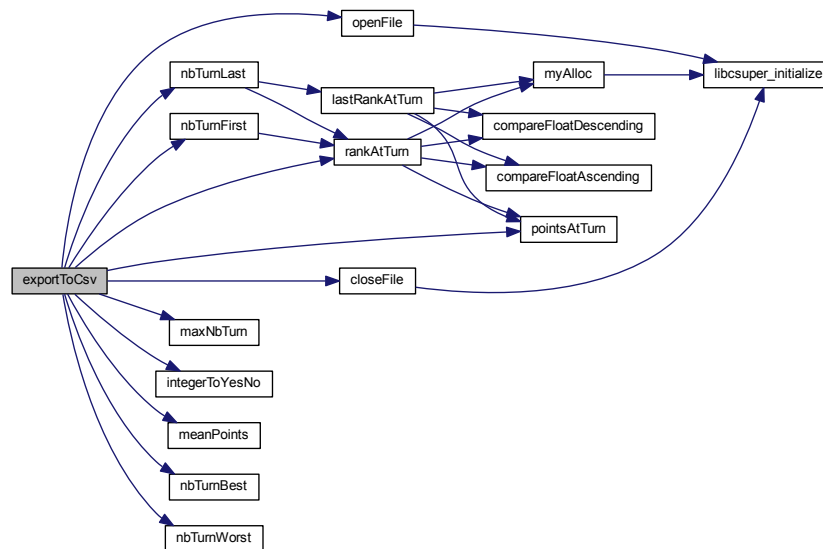
Parameters

| | | |
|----|-----------------------|--|
| in | <i>ptr_csu_struct</i> | a pointer on a csuStruct |
| in | <i>filename</i> | the filename |

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:



4.8.4.11 bool exportToM (csuStruct * ptr_csu_struct, char * filename)

Export the csu structure to a m file (octave/matlab file)

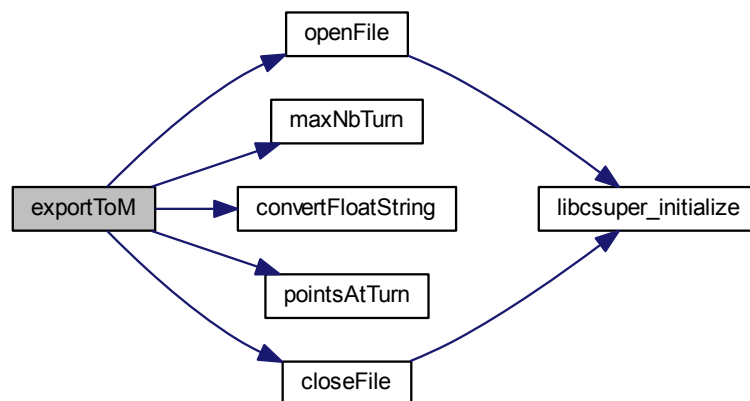
Parameters

| | | |
|----|-----------------------|--|
| in | <i>ptr_csu_struct</i> | a pointer on a csuStruct |
| in | <i>filename</i> | the filename |

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:

**4.8.4.12** `bool exportToPdf (csuStruct * ptr_csu_struct, char * filename)`

Export the csu structure to a pdf

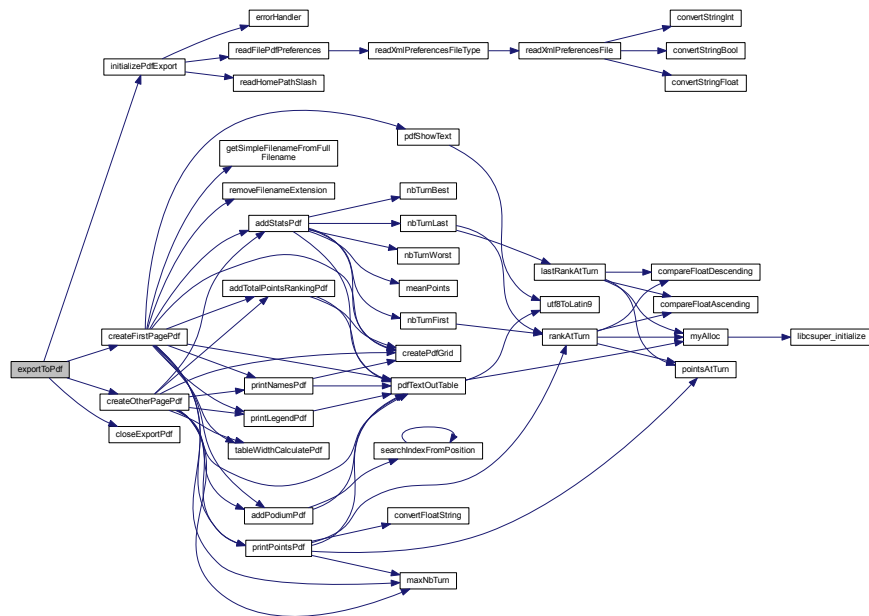
Parameters

| | | |
|----|-----------------------|--|
| in | <i>ptr_csu_struct</i> | a pointer on a csuStruct |
| in | <i>filename</i> | the filename of the pdf file |

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:



4.8.4.13 bool initializePdfExport (export_pdf * ptr_export_pdf, csuStruct * ptr_csu_struct)

Initialize the exportation

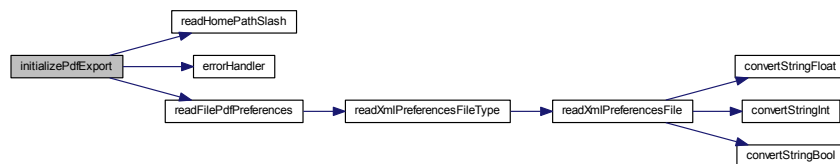
Parameters

| | | |
|----|----------------|---|
| in | ptr_csu_struct | a pointer on a csuStruct |
| in | ptr_export_pdf | a pointer on a export_pdf |

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:



4.8.4.14 void pdfShowText (HPDF_Page page, char * text, export_pdf * ptr_export_pdf)

Print the text in the page

Parameters

| | | |
|----|-----------------------|---|
| in | <i>page</i> | the page |
| in | <i>text</i> | the text to print |
| in | <i>ptr_export_pdf</i> | a pointer on a export_pdf |

Here is the call graph for this function:



4.8.4.15 `bool pdfTextOutTable (HPDF_Page page, float pos_min_x, float pos_y, const char * text, float max_width, int ranking, export_pdf * ptr_export_pdf)`

Print the text in the page center with a maximum width

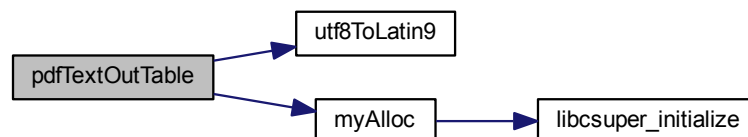
Parameters

| | | |
|----|-----------------------|---|
| in | <i>page</i> | the page |
| in | <i>pos_min_x</i> | the minimum position on the x axis |
| in | <i>pos_y</i> | the position on the y axis |
| in | <i>text</i> | the text to print |
| in | <i>max_width</i> | the maximum width of the text |
| in | <i>ranking</i> | the ranking of the player to determine the color of the text, put 0 to print in black |
| in | <i>ptr_export_pdf</i> | a pointer on a export_pdf |

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:



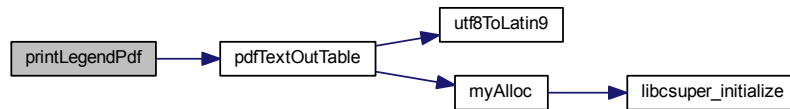
4.8.4.16 `void printLegendPdf (export_pdf * ptr_export_pdf, csuStruct * ptr_csu_struct, float * pos_y, float table_width, HPDF_Page page)`

Print the legend on a pdf page if needed

Parameters

| | | |
|----|-----------------------|---|
| in | <i>page</i> | the page |
| in | <i>pos_y</i> | a pointer to the first position on the y axis |
| in | <i>table_width</i> | the width of a table |
| in | <i>ptr_csu_struct</i> | a pointer on a csuStruct |
| in | <i>ptr_export_pdf</i> | a pointer on a export_pdf |

Here is the call graph for this function:



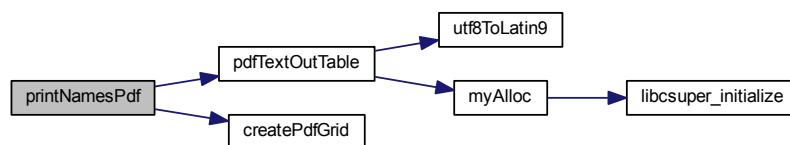
4.8.4.17 void printNamesPdf ([export_pdf](#) * *ptr_export_pdf*, [csuStruct](#) * *ptr_csu_struct*, float * *pos_y*, float *table_width*, [HPDF_Page](#) *page*)

Print the names on a pdf page

Parameters

| | | |
|----|-----------------------|---|
| in | <i>page</i> | the page |
| in | <i>pos_y</i> | a pointer to the first position on the y axis |
| in | <i>table_width</i> | the width of a table |
| in | <i>ptr_csu_struct</i> | a pointer on a csuStruct |
| in | <i>ptr_export_pdf</i> | a pointer on a export_pdf |

Here is the call graph for this function:



4.8.4.18 void printPointsPdf ([HPDF_Page](#) *page*, float * *pos_y*, [csuStruct](#) * *ptr_csu_struct*, [export_pdf](#) * *ptr_export_pdf*, float *table_width*)

Print the points on a pdf page

Parameters

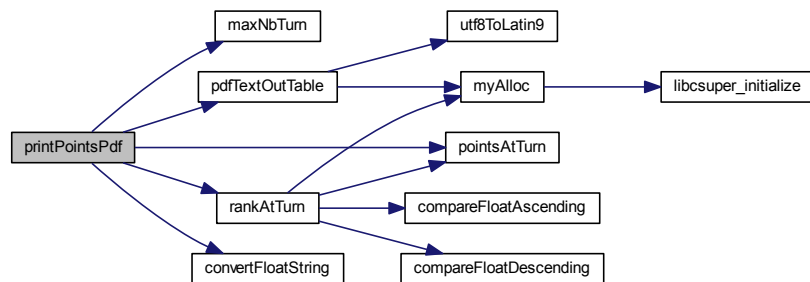
| | | |
|----|-------------|----------|
| in | <i>page</i> | the page |
|----|-------------|----------|

| | | |
|----|-----------------------|---|
| in | <i>pos_y</i> | a pointer to the first position on the y axis |
| in | <i>table_width</i> | the width of a table |
| in | <i>ptr_csu_struct</i> | a pointer on a csuStruct |
| in | <i>ptr_export_pdf</i> | a pointer on a export_pdf |

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:



4.8.4.19 float tableWidthCalculatePdf (export_pdf * ptr_export_pdf, csuStruct * ptr_csu_struct, HPDF_Page page)

Calculate the table width

Parameters

| | | |
|----|-----------------------|---|
| in | <i>page</i> | the page |
| in | <i>ptr_csu_struct</i> | a pointer on a csuStruct |
| in | <i>ptr_export_pdf</i> | a pointer on a export_pdf |

Returns

the table width

4.9 file.c File Reference

Files function of libcsuper.

```
#include "file.h"
```

Functions

- FILE * [openFile](#) (char file_name[], char mode[])
- int [closeFile](#) (FILE *ptr_file)
- int [readFileSize](#) (FILE *ptr_file)
- bool [deleteFile](#) (char *file_name)
- bool [renameFile](#) (char *old_name, char *new_name)

4.9.1 Detailed Description

Files function of libcsuper.

Author

Remi BERTHO

Date

05/07/14

Version

4.0.1

4.9.2 Function Documentation

4.9.2.1 `int closeFile (FILE * ptr_file)`

Close the file

Parameters

| | | |
|-----------------|------------------------|----------|
| <code>in</code> | <code>*ptr_file</code> | the file |
|-----------------|------------------------|----------|

Returns

0 if everything is OK, 1 otherwise

Here is the call graph for this function:



4.9.2.2 `bool deleteFile (char * file_name)`

Delete a file

Parameters

| | | |
|-----------------|-------------------------|--------------|
| <code>in</code> | <code>*file_name</code> | the filename |
|-----------------|-------------------------|--------------|

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:

**4.9.2.3 FILE * openFile (char file_name[], char mode[])**

Open a file with his name and with a specific mode.

Parameters

| | | |
|----|--------------------|--------------|
| in | <i>file_name[]</i> | the filename |
| in | <i>mode[]</i> | the mode |

Returns

a pointer to the open file, NULL if there was a problem

Here is the call graph for this function:

**4.9.2.4 int readFileSize (FILE * ptr_file)**

Read the size of the file

Parameters

| | | |
|----|------------------|----------|
| in | <i>*ptr_file</i> | the file |
|----|------------------|----------|

Returns

the size of the file

4.9.2.5 bool renameFile (char * old_name, char * new_name)

Rename a file.

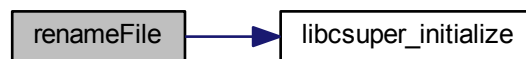
Parameters

| | | |
|----|------------------|--------------------------|
| in | <i>*old_name</i> | the old name of the file |
| in | <i>*new_name</i> | the new name of the file |

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:



4.10 file.h File Reference

Header for the files function of libcsuper.

```
#include "share.h"
```

Functions

- FILE * [openFile](#) (char nome[], char mode[])
- int [closeFile](#) (FILE *ptr_file)
- int [readFileSize](#) (FILE *ptr_file)
- bool [deleteFile](#) (char *file_name)
- bool [renameFile](#) (char *old_name, char *new_name)

4.10.1 Detailed Description

Header for the files function of libcsuper.

Author

Remi BERTHO

Date

05/07/14

Version

4.0.1

4.10.2 Function Documentation

4.10.2.1 int closeFile (FILE * ptr_file)

Close the file

Parameters

| | | |
|----|------------------|----------|
| in | <i>*ptr_file</i> | the file |
|----|------------------|----------|

Returns

0 if everything is OK, 1 otherwise

Here is the call graph for this function:



4.10.2.2 bool deleteFile (char * file_name)

Delete a file

Parameters

| | | |
|----|-------------------|--------------|
| in | <i>*file_name</i> | the filename |
|----|-------------------|--------------|

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:



4.10.2.3 FILE* openFile (char file_name[], char mode[])

Open a file with his name and with a specific mode.

Parameters

| | | |
|----|--------------------|--------------|
| in | <i>file_name[]</i> | the filename |
|----|--------------------|--------------|

| | | |
|----|---------------|----------|
| in | <i>mode[]</i> | the mode |
|----|---------------|----------|

Returns

a pointer to the open file, NULL if there was a problem

Here is the call graph for this function:

**4.10.2.4 int readFileSize (FILE * *ptr_file*)**

Read the size of the file

Parameters

| | | |
|----|------------------|----------|
| in | <i>*ptr_file</i> | the file |
|----|------------------|----------|

Returns

the size of the file

4.10.2.5 bool renameFile (char * *old_name*, char * *new_name*)

Rename a file.

Parameters

| | | |
|----|------------------|--------------------------|
| in | <i>*old_name</i> | the old name of the file |
| in | <i>*new_name</i> | the new name of the file |

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:



4.11 filename.c File Reference

Essential function of libcsuper.

```
#include "filename.h"
```

Functions

- void [addFileExtension](#) (char *file_name, char *extension)
- void [addFileCsuExtension](#) (char *file_name)
- void [addFilePdfExtension](#) (char *file_name)
- void [addFileCsvExtension](#) (char *file_name)
- void [addFileGnuplotExtension](#) (char *file_name)
- void [addFileDatExtension](#) (char *file_name)
- void [removeFileExtension](#) (char *file_name)
- bool [getFolderFromFilename](#) (char *file_name_to_folder)
- bool [getSimpleFilenameFromFullFilename](#) (char *full_filename, char *simple_filename)
- bool [checkPath](#) (char *path)
- bool [checkFilename](#) (char *filename, char *folder)
- void [readHomePath](#) (char *path)
- void [readHomePathSlash](#) (char *path)
- bool [removeFilenameExtension](#) (char *filename)

4.11.1 Detailed Description

Essential function of libcsuper.

Author

Remi BERTHO

Date

22/12/14

Version

4.1.0

4.11.2 Function Documentation

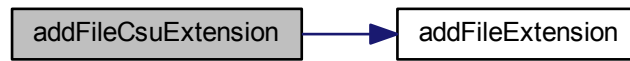
4.11.2.1 void addFileCsuExtension (char * *file_name*)

Add the csu file extension

Parameters

| | | |
|----|------------------|--------------|
| in | <i>file_name</i> | the filename |
|----|------------------|--------------|

Here is the call graph for this function:



4.11.2.2 void addFileCsvExtension (char * *file_name*)

Add the csv file extension

Parameters

| | | |
|----|------------------|--------------|
| in | <i>file_name</i> | the filename |
|----|------------------|--------------|

Here is the call graph for this function:



4.11.2.3 void addFileDatExtension (char * *file_name*)

Add the dat file extension

Parameters

| | | |
|----|------------------|--------------|
| in | <i>file_name</i> | the filename |
|----|------------------|--------------|

Here is the call graph for this function:



4.11.2.4 void addFileExtension (char * *file_name*, char * *extension*)

Add the a file extension to a filename

Parameters

| | | |
|----|------------------|---------------|
| in | <i>file_name</i> | the filename |
| in | <i>extension</i> | the extension |

4.11.2.5 void addFileGnuplotExtension (char * *file_name*)

Add the gnuplot file extension

Parameters

| | | |
|----|------------------|--------------|
| in | <i>file_name</i> | the filename |
|----|------------------|--------------|

Here is the call graph for this function:

4.11.2.6 void addFilePdfExtension (char * *file_name*)

Add the pdf file extension

Parameters

| | | |
|----|------------------|--------------|
| in | <i>file_name</i> | the filename |
|----|------------------|--------------|

Here is the call graph for this function:

4.11.2.7 bool checkFilename (char * *filename*, char * *folder*)

Test if the filename is valid

Parameters

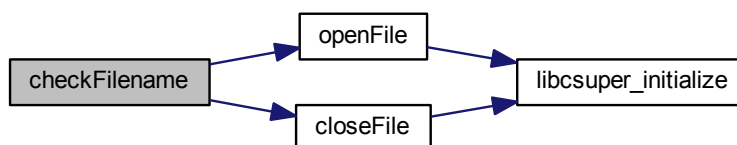
| | | |
|---------|-------------------|--------------|
| in, out | * <i>filename</i> | the filename |
|---------|-------------------|--------------|

| | | |
|----------------|----------------|---|
| <i>in, out</i> | <i>*folder</i> | the folder where the filename will be tested, may be "" |
|----------------|----------------|---|

Returns

true if the filename is valid OK, false otherwise

Here is the call graph for this function:

**4.11.2.8 bool checkPath (char * path)**

Test if the path is valid

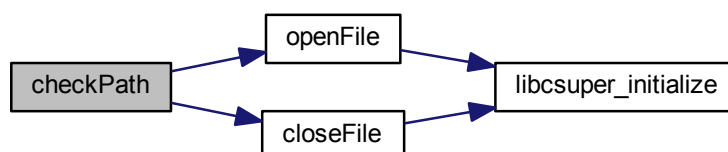
Parameters

| | | |
|----------------|--------------|----------|
| <i>in, out</i> | <i>*path</i> | the path |
|----------------|--------------|----------|

Returns

true if the path is valid OK, false otherwise

Here is the call graph for this function:

**4.11.2.9 bool getFolderFromFilename (char * file_name_to_folder)**

Transform a filename into his folder

Parameters

| | | |
|----|----------------------------------|--------------|
| in | <i>file_name_to_↔ folder</i> | the filename |
|----|----------------------------------|--------------|

Returns

true if everything is OK, false otherwise

4.11.2.10 bool getSimpleFilenameFromFullFilename (char * *full_filename*, char * *simple_filename*)

Transform a full filename into his simple filename (without the folder)

Parameters

| | | |
|----|------------------------|-------------------|
| in | <i>full_filename</i> | the full filename |
| in | <i>simple_filename</i> | the full filename |

Returns

true if everything is OK, false otherwise

4.11.2.11 void readHomePath (char * *path*)

Read the home path

Parameters

| | | |
|---------|-------------|----------|
| in, out | <i>path</i> | the path |
|---------|-------------|----------|

Read the home path with a slash at the end

Parameters

| | | |
|---------|-------------|----------|
| in, out | <i>path</i> | the path |
|---------|-------------|----------|

4.11.2.12 void readHomePathSlash (char * *path*)4.11.2.13 void removeFileExtension (char * *file_name*)

Remove the file extension file extension

Parameters

| | | |
|----|------------------|--------------|
| in | <i>file_name</i> | the filename |
|----|------------------|--------------|

4.11.2.14 bool removeFilenameExtension (char * *filename*)

Remove the file extension of the filename

Parameters

| | | |
|----|-----------------|--------------|
| in | <i>filename</i> | the filename |
|----|-----------------|--------------|

Returns

true if everything is OK, false otherwise

4.12 filename.h File Reference

Header for the essential function of libcsuper.

```
#include "preferences_files.h"
```

Functions

- void [addFileExtension](#) (char *file_name, char *extension)
- void [addFileCsuExtension](#) (char *file_name)
- void [addFilePdfExtension](#) (char *file_name)
- void [addFileCsvExtension](#) (char *file_name)
- void [addFileGnuplotExtension](#) (char *file_name)
- void [addFileDatExtension](#) (char *file_name)
- void [removeFileExtension](#) (char *file_name)
- bool [getFolderFromFilename](#) (char *file_name_to_folder)
- bool [getSimpleFilenameFromFullFilename](#) (char *full_filename, char *simple_filename)
- bool [checkPath](#) (char *path)
- bool [checkFilename](#) (char *filename, char *folder)
- void [readHomePath](#) (char *path)
- void [readHomePathSlash](#) (char *path)
- bool [removeFilenameExtension](#) (char *filename)

4.12.1 Detailed Description

Header for the essential function of libcsuper.

Author

Remi BERTHO

Date

22/12/14

Version

4.1.0

4.12.2 Function Documentation

4.12.2.1 void addFileCsuExtension (char * *file_name*)

Add the csu file extension

Parameters

| | | |
|----|------------------|--------------|
| in | <i>file_name</i> | the filename |
|----|------------------|--------------|

Here is the call graph for this function:



4.12.2.2 void addFileCsvExtension (char * *file_name*)

Add the csv file extension

Parameters

| | | |
|----|------------------|--------------|
| in | <i>file_name</i> | the filename |
|----|------------------|--------------|

Here is the call graph for this function:



4.12.2.3 void addFileDatExtension (char * *file_name*)

Add the dat file extension

Parameters

| | | |
|----|------------------|--------------|
| in | <i>file_name</i> | the filename |
|----|------------------|--------------|

Here is the call graph for this function:



4.12.2.4 void addFileExtension (char * *file_name*, char * *extension*)

Add the a file extension to a filename

Parameters

| | | |
|----|------------------|---------------|
| in | <i>file_name</i> | the filename |
| in | <i>extension</i> | the extension |

4.12.2.5 void addFileGnuplotExtension (char * *file_name*)

Add the gnuplot file extension

Parameters

| | | |
|----|------------------|--------------|
| in | <i>file_name</i> | the filename |
|----|------------------|--------------|

Here is the call graph for this function:

4.12.2.6 void addFilePdfExtension (char * *file_name*)

Add the pdf file extension

Parameters

| | | |
|----|------------------|--------------|
| in | <i>file_name</i> | the filename |
|----|------------------|--------------|

Here is the call graph for this function:

4.12.2.7 bool checkFilename (char * *filename*, char * *folder*)

Test if the filename is valid

Parameters

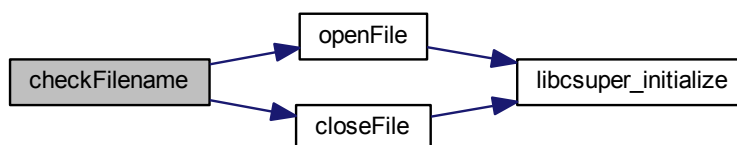
| | | |
|---------|-------------------|--------------|
| in, out | * <i>filename</i> | the filename |
|---------|-------------------|--------------|

| | | |
|----------------|----------------|---|
| <i>in, out</i> | <i>*folder</i> | the folder where the filename will be tested, may be "" |
|----------------|----------------|---|

Returns

true if the filename is valid OK, false otherwise

Here is the call graph for this function:

**4.12.2.8 bool checkPath (char * path)**

Test if the path is valid

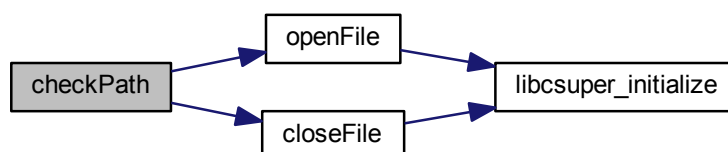
Parameters

| | | |
|----------------|--------------|----------|
| <i>in, out</i> | <i>*path</i> | the path |
|----------------|--------------|----------|

Returns

true if the path is valid OK, false otherwise

Here is the call graph for this function:

**4.12.2.9 bool getFolderFromFilename (char * file_name_to_folder)**

Transform a filename into his folder

Parameters

| | | |
|----|----------------------------------|--------------|
| in | <i>file_name_to_↔ folder</i> | the filename |
|----|----------------------------------|--------------|

Returns

true if everything is OK, false otherwise

4.12.2.10 bool getSimpleFilenameFromFullFilename (char * *full_filename*, char * *simple_filename*)

Transform a full filename into his simple filename (without the folder)

Parameters

| | | |
|----|------------------------|-------------------|
| in | <i>full_filename</i> | the full filename |
| in | <i>simple_filename</i> | the full filename |

Returns

true if everything is OK, false otherwise

4.12.2.11 void readHomePath (char * *path*)

Read the home path

Parameters

| | | |
|---------|-------------|----------|
| in, out | <i>path</i> | the path |
|---------|-------------|----------|

Read the home path with a slash at the end

Parameters

| | | |
|---------|-------------|----------|
| in, out | <i>path</i> | the path |
|---------|-------------|----------|

4.12.2.12 void readHomePathSlash (char * *path*)**4.12.2.13 void removeFileExtension (char * *file_name*)**

Remove the file extension file extension

Parameters

| | | |
|----|------------------|--------------|
| in | <i>file_name</i> | the filename |
|----|------------------|--------------|

4.12.2.14 bool removeFilenameExtension (char * *filename*)

Remove the file extension of the filename

Parameters

| | | |
|----|-----------------|--------------|
| in | <i>filename</i> | the filename |
|----|-----------------|--------------|

Returns

true if everything is OK, false otherwise

4.13 game_config.c File Reference

Game configuration.

```
#include "game_config.h"
```

Functions

- [list_game_config](#) * [newListGameConfig](#) (int nb_config)
- void [closeListGameConfig](#) ([list_game_config](#) *ptr_list_config)
- void [addConfigListGameConfig](#) ([list_game_config](#) *ptr_list_config, [game_config](#) config)
- void [removeConfigListGameConfig](#) ([list_game_config](#) *ptr_list_config, [game_config](#) config)
- bool [makeConfigListFile](#) (char *home_path)
- [list_game_config](#) * [readConfigListFile](#) (char *home_path)
- bool [addConfigListFile](#) (char *new_config_name, char *home_path)
- bool [removeConfigListFile](#) (int index_delete, [list_game_config](#) *ptr_list_config, char *home_path)
- bool [newConfigFile](#) ([game_config](#) config, char *home_path)
- bool [removeConfigFile](#) (char *config_name, char *home_path)
- bool [readConfigFile](#) (int index_read, [list_game_config](#) *ptr_list_config, [game_config](#) *ptr_config, char *home_path)
- bool [exportConfigFile](#) (char *home_path, char *file_name, int *id, int nb_id)
- bool [importConfigFile](#) (char *home_path, char *file_name, int *id, int nb_id)
- [list_game_config](#) * [newListGameConfigFromImport](#) (char *filename)
- bool [writeXmlListGameConfig](#) (char *filename, [list_game_config](#) *ptr_list_config)
- [list_game_config](#) * [readXmlListGameConfig](#) (char *filename)
- bool [writeXmlListGameConfigWithId](#) (char *filename, [list_game_config](#) *ptr_list_config, int *id, int nb_id)
- [list_game_config](#) * [readXmlListGameConfigWithId](#) (char *filename, int *id, int nb_id)

4.13.1 Detailed Description

Game configuration.

Author

Remi BERTHO

Date

29/04/14

Version

2.4.0

4.13.2 Function Documentation

4.13.2.1 bool addConfigListFile (char * new_config_name, char * home_path)

Deprecated, do nothing

Parameters

| | | |
|----|------------------------|--|
| in | <i>new_config_name</i> | the name of the new game configuration |
| in | <i>home_path</i> | the path to the home directory |

Returns

true

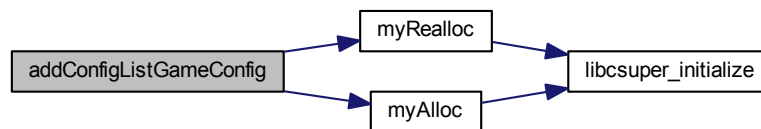
4.13.2.2 void addConfigListGameConfig (list_game_config * ptr_list_config, game_config config)

Add a config to the config list

Parameters

| | | |
|----|-------------------------|---|
| in | <i>*ptr_list_config</i> | a pointer on a list of game configuration |
| in | <i>config</i> | a game configuration |

Here is the call graph for this function:



4.13.2.3 void closeListGameConfig (list_game_config * ptr_list_config)

Free a list of game configuration

Parameters

| | | |
|----|-------------------------|---|
| in | <i>*ptr_list_config</i> | a pointer on a list of game configuration |
|----|-------------------------|---|

4.13.2.4 bool exportConfigFile (char * home_path, char * file_name, int * id, int nb_id)

Export all config file into a file.

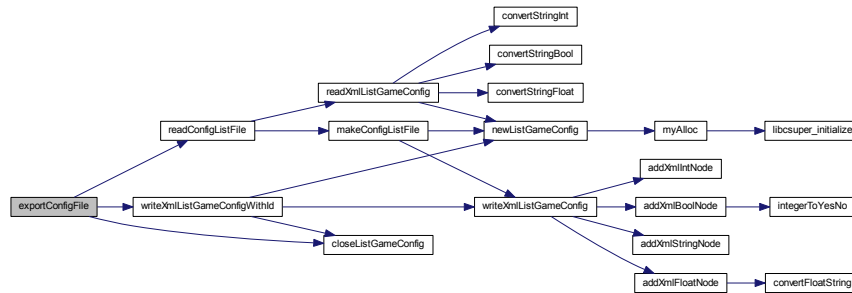
Parameters

| | | |
|----|------------------|--|
| in | <i>file_name</i> | the filename of the exported file. |
| in | <i>home_path</i> | the path to the home directory |
| in | <i>id</i> | the ids of the game configuration which will be exported |
| in | <i>nb_id</i> | the number of game configuration which will be exported |

Returns

true if there is no problem, false otherwise

Here is the call graph for this function:



4.13.2.5 bool importConfigFile (char * home_path, char * file_name, int * id, int nb_id)

Import all config file from a file.

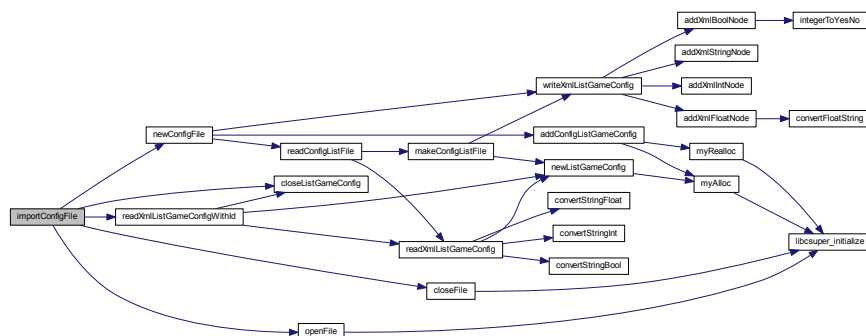
Parameters

| | | |
|----|------------------|---|
| in | <i>file_name</i> | the filename of the imported file. |
| in | <i>home_path</i> | the path to the home directory |
| in | <i>id</i> | the id of the game configuration which will be imported |
| in | <i>nb_id</i> | the number of game configuration which will be imported |

Returns

true if there is no problem, false otherwise

Here is the call graph for this function:



4.13.2.6 bool makeConfigListFile (char * home_path)

Create the files which contain the list of games configurations

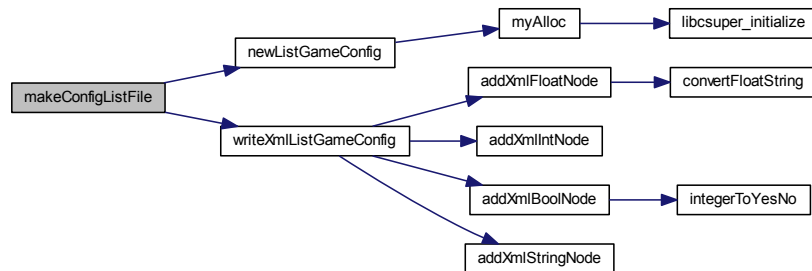
Parameters

| | | |
|----|-------------------|--------------------------------|
| in | <i>*home_path</i> | the path to the home directory |
|----|-------------------|--------------------------------|

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:



4.13.2.7 bool newConfigFile (game_config config, char * home_path)

Save a new game configuration file

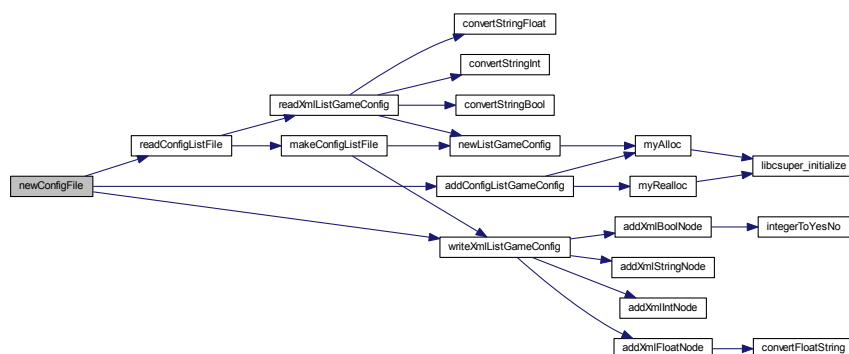
Parameters

| | | |
|----|------------------|--------------------------------|
| in | <i>config</i> | the gale configuration |
| in | <i>home_path</i> | the path to the home directory |

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:



4.13.2.8 list_game_config * newListGameConfig (int nb_config)

Create a list of game configuration.

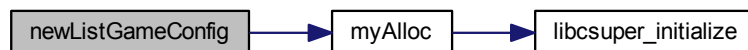
Parameters

| | | |
|----|------------------|----------------------------------|
| in | <i>nb_config</i> | the number of game configuration |
|----|------------------|----------------------------------|

Returns

une [list_game_config](#)

Here is the call graph for this function:

4.13.2.9 `list_game_config * newListGameConfigFromImport (char * filename)`

Create a [list_game_config](#) with a import file

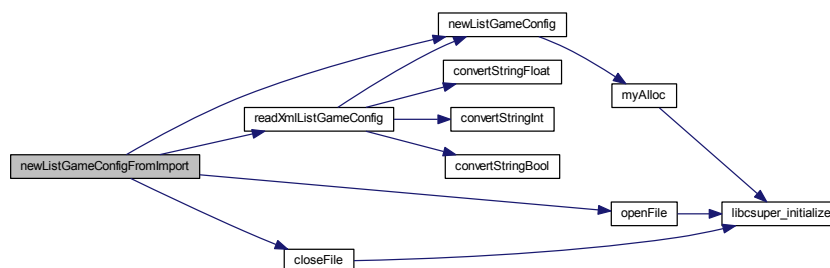
Parameters

| | | |
|----|-----------------|------------------------------------|
| in | <i>filename</i> | the filename of the exported file. |
|----|-----------------|------------------------------------|

Returns

a [list_game_config](#)

Here is the call graph for this function:

4.13.2.10 `bool readConfigFile (int index_read, list_game_config * ptr_list_config, game_config * ptr_config, char * home_path)`

Read a game configuration file.

Parameters

| | | |
|----|------------------------|--|
| in | <i>index_read</i> | the index of the game configuration to be read |
| in | <i>ptr_list_config</i> | a pointer on the game configuration list |
| in | <i>ptr_config</i> | a pointer on a game configuration |
| in | <i>home_path</i> | the path to the home directory |

Returns

a [list_game_config](#)

4.13.2.11 `list_game_config * readConfigListFile (char * home_path)`

Read the file which contain the list of game configuration

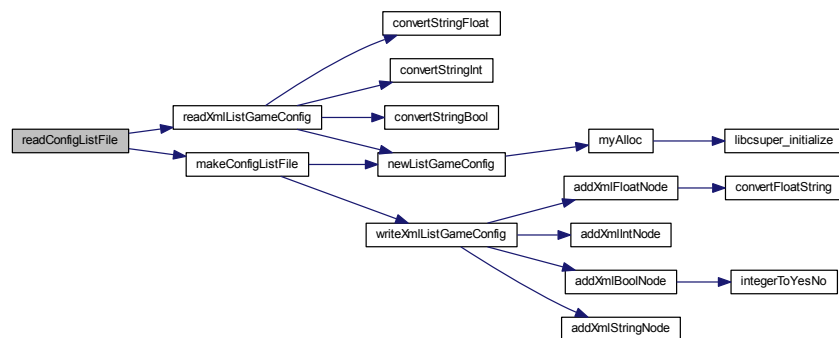
Parameters

| | | |
|----|-------------------|--------------------------------|
| in | <i>*home_path</i> | the path to the home directory |
|----|-------------------|--------------------------------|

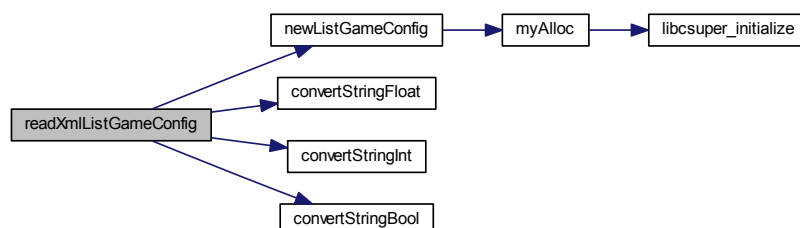
Returns

a [list_game_config](#)

Here is the call graph for this function:

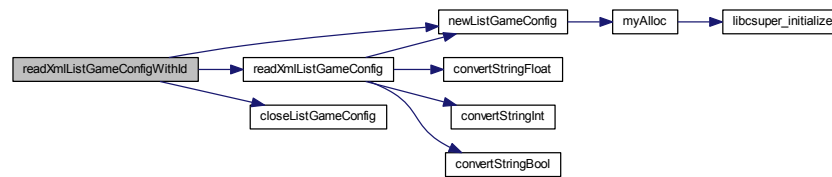
4.13.2.12 `list_game_config* readXmlListGameConfig (char * filename)`

Here is the call graph for this function:



4.13.2.13 `list_game_config* readXmlListGameConfigWithId (char * filename, int * id, int nb_id)`

Here is the call graph for this function:

4.13.2.14 `bool removeConfigFile (char * config_name, char * home_path)`

Deprecated, do nothing

Parameters

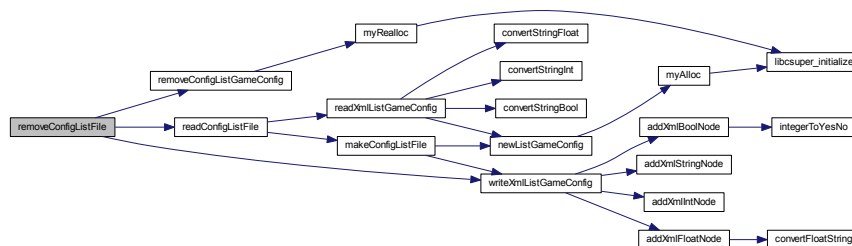
| | | |
|----|--------------------|--|
| in | <i>config_name</i> | the name of the game configuration which will be deleted |
| in | <i>home_path</i> | the path to the home directory |

Returns

true

4.13.2.15 `bool removeConfigListFile (int index_delete, list_game_config * ptr_list_config, char * home_path)`

Here is the call graph for this function:

4.13.2.16 `void removeConfigListGameConfig (list_game_config * ptr_list_config, game_config config)`

Remove a config to the config list

Parameters

| | | |
|----|-------------------------|---|
| in | <i>*ptr_list_config</i> | a pointer on a list of game configuration |
|----|-------------------------|---|

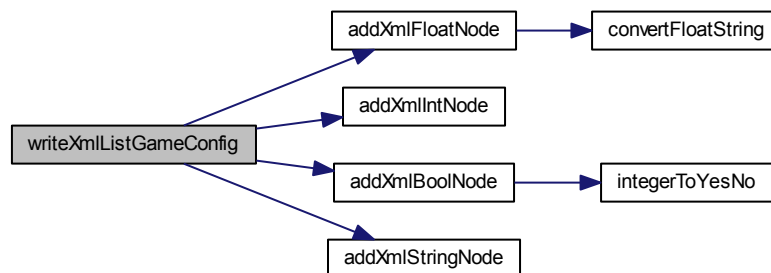
| | | |
|-----------------|---------------------|----------------------|
| <code>in</code> | <code>config</code> | a game configuration |
|-----------------|---------------------|----------------------|

Here is the call graph for this function:



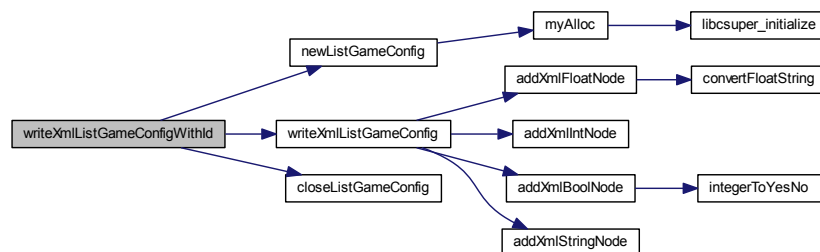
4.13.2.17 `bool writeXmlListGameConfig (char * filename, list_game_config * ptr_list_config)`

Here is the call graph for this function:



4.13.2.18 `bool writeXmlListGameConfigWithId (char * filename, list_game_config * ptr_list_config, int * id, int nb_id)`

Here is the call graph for this function:



4.14 `game_config.h` File Reference

Game configurations.


```
#include <math.h>
#include "csu_struct.h"
#include "preferences_files.h"
```

Data Structures

- struct [list_game_config](#)

Macros

- #define [CONFIGURATION_FOLDER_NAME](#) "config"
- #define [CONFIGURATION_FILE_NAME](#) "configuration"
- #define [CONFIGURATION_XML_FILENAME](#) "game_configuration.xml"
- #define [STRING_CHECK_GAME_CONFIG](#) "Csuper_Game_Configuration"
- #define [GAME_CONFIG_FILE_XML_VERSION](#) 1.0

Functions

- [list_game_config](#) * [newListGameConfig](#) (int nb_config)
- void [closeListGameConfig](#) ([list_game_config](#) *ptr_list_config)
- void [addConfigListGameConfig](#) ([list_game_config](#) *ptr_list_config, [game_config](#) config)
- void [removeConfigListGameConfig](#) ([list_game_config](#) *ptr_list_config, [game_config](#) config)
- bool [makeConfigListFile](#) (char *home_path)
- [list_game_config](#) * [readConfigListFile](#) (char *home_path)
- bool [addConfigListFile](#) (char *new_config_name, char *home_path)
- bool [removeConfigListFile](#) (int index_delete, [list_game_config](#) *ptr_list_config, char *home_path)
- bool [newConfigFile](#) ([game_config](#) config, char *home_path)
- bool [removeConfigFile](#) (char *config_name, char *home_path)
- bool [readConfigFile](#) (int index_read, [list_game_config](#) *ptr_list_config, [game_config](#) *ptr_config, char *home_path)
- bool [exportConfigFile](#) (char *home_path, char *file_name, int *id, int nb_id)
- bool [importConfigFile](#) (char *home_path, char *file_name, int *id, int nb_id)
- [list_game_config](#) * [newListGameConfigFromImport](#) (char *filename)
- [list_game_config](#) * [readXmlListGameConfig](#) (char *filename)
- bool [writeXmlListGameConfig](#) (char *filename, [list_game_config](#) *ptr_list_config)
- [list_game_config](#) * [readXmlListGameConfigWithId](#) (char *filename, int *id, int nb_id)
- bool [writeXmlListGameConfigWithId](#) (char *filename, [list_game_config](#) *ptr_list_config, int *id, int nb_id)

4.14.1 Detailed Description

Game configurations.

Author

Remi BERTHO

Date

29/04/14

Version

2.4.0

4.14.2 Macro Definition Documentation

4.14.2.1 `#define CONFIGURATION_FILE_NAME "configuration"`

Define the name of the file which contain the list of the game configurations

4.14.2.2 `#define CONFIGURATION_FOLDER_NAME "config"`

Define the name of the folder which contain the game configurations

4.14.2.3 `#define CONFIGURATION_XML_FILENAME "game_configuration.xml"`

Define the filename of the file which contain the game configuration

4.14.2.4 `#define GAME_CONFIG_FILE_XML_VERSION 1.0`

Define the version of the XML game configuration file

4.14.2.5 `#define STRING_CHECK_GAME_CONFIG "Csuper_Game_Configuration"`

String for checking if the file is game configuration file.

4.14.3 Function Documentation

4.14.3.1 `bool addConfigListFile (char * new_config_name, char * home_path)`

Depreciated, do nothing

Parameters

| | | |
|----|------------------------|--|
| in | <i>new_config_name</i> | the name of the new game configuration |
| in | <i>home_path</i> | the path to the home directory |

Returns

true

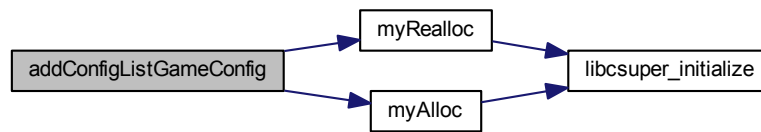
4.14.3.2 `void addConfigListGameConfig (list_game_config * ptr_list_config, game_config config)`

Add a config to the config list

Parameters

| | | |
|----|-------------------------|---|
| in | <i>*ptr_list_config</i> | a pointer on a list of game configuration |
| in | <i>config</i> | a game configuration |

Here is the call graph for this function:



4.14.3.3 void closeListGameConfig (list_game_config * ptr_list_config)

Free a list of game configuration

Parameters

| | | |
|----|-------------------------|---|
| in | <i>*ptr_list_config</i> | a pointer on a list of game configuration |
|----|-------------------------|---|

4.14.3.4 bool exportConfigFile (char * home_path, char * file_name, int * id, int nb_id)

Export all config file into a file.

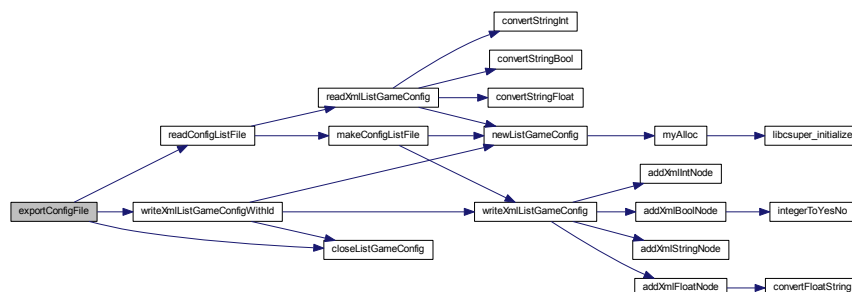
Parameters

| | | |
|----|------------------|--|
| in | <i>file_name</i> | the filename of the exported file. |
| in | <i>home_path</i> | the path to the home directory |
| in | <i>id</i> | the ids of the game configuration which will be exported |
| in | <i>nb_id</i> | the number of game configuration which will be exported |

Returns

true if there is no problem, false otherwise

Here is the call graph for this function:



4.14.3.5 bool importConfigFile (char * home_path, char * file_name, int * id, int nb_id)

Import all config file from a file.

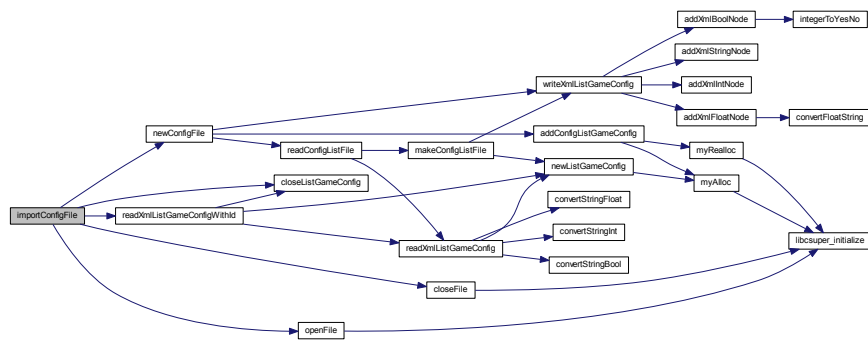
Parameters

| | | |
|----|------------------|---|
| in | <i>file_name</i> | the filename of the imported file. |
| in | <i>home_path</i> | the path to the home directory |
| in | <i>id</i> | the id of the game configuration which will be imported |
| in | <i>nb_id</i> | the number of game configuration which will be imported |

Returns

true if there is no problem, false otherwise

Here is the call graph for this function:



4.14.3.6 bool makeConfigListFile (char * home_path)

Create the files which contain the list of games configurations

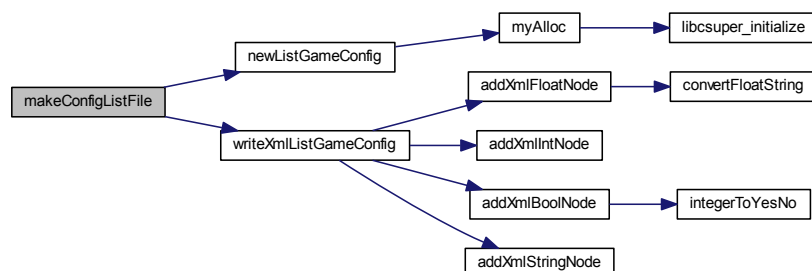
Parameters

| | | |
|----|-------------------|--------------------------------|
| in | <i>*home_path</i> | the path to the home directory |
|----|-------------------|--------------------------------|

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:



4.14.3.7 `bool newConfigFile (game_config config, char * home_path)`

Save a new game configuration file

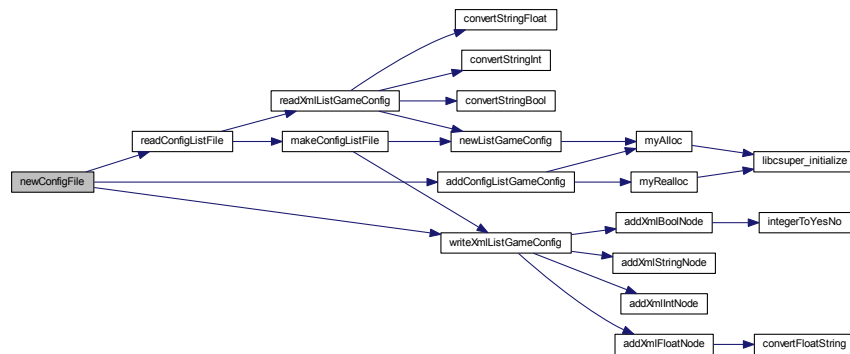
Parameters

| | | |
|----|------------------|--------------------------------|
| in | <i>config</i> | the gale configuration |
| in | <i>home_path</i> | the path to the home directory |

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:



4.14.3.8 list_game_config* newListGameConfig (int nb_config)

Create a list of game configuration.

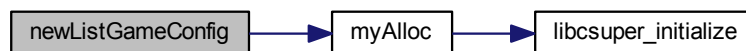
Parameters

| | | |
|----|------------------|----------------------------------|
| in | <i>nb_config</i> | the number of game configuration |
|----|------------------|----------------------------------|

Returns

une [list_game_config](#)

Here is the call graph for this function:



4.14.3.9 list_game_config* newListGameConfigFromImport (char * filename)

Create a [list_game_config](#) with a import file

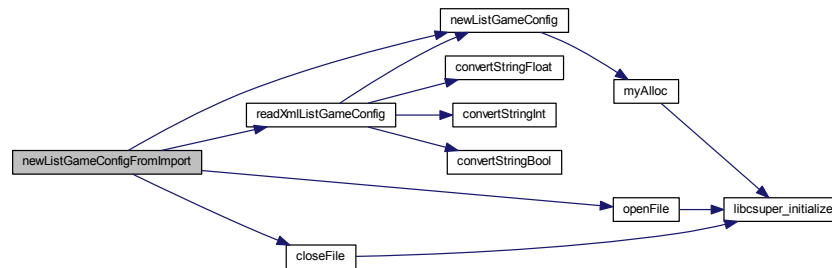
Parameters

| | | |
|----|-----------------|------------------------------------|
| in | <i>filename</i> | the filename of the exported file. |
|----|-----------------|------------------------------------|

Returns

a [list_game_config](#)

Here is the call graph for this function:



4.14.3.10 bool readConfigFile (int *index_read*, list_game_config * *ptr_list_config*, game_config * *ptr_config*, char * *home_path*)

Read a game configuration file.

Parameters

| | | |
|----|------------------------|--|
| in | <i>index_read</i> | the index of the game configuration to be read |
| in | <i>ptr_list_config</i> | a pointer on the game configuration list |
| in | <i>ptr_config</i> | a pointer on a game configuration |
| in | <i>home_path</i> | the path to the home directory |

Returns

a [list_game_config](#)

4.14.3.11 list_game_config* readConfigListFile (char * *home_path*)

Read the file which contain the list of game configuration

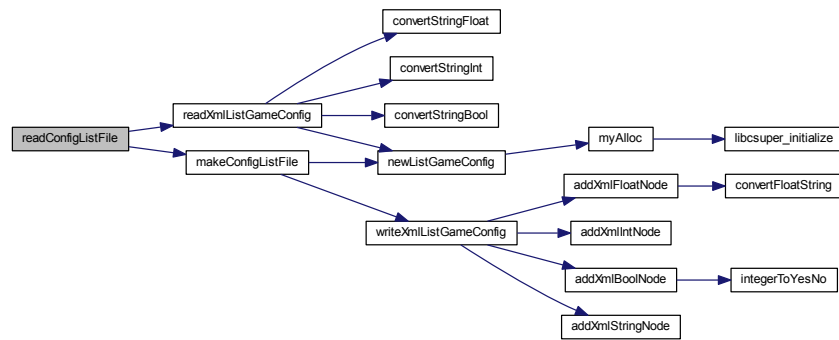
Parameters

| | | |
|----|-------------------|--------------------------------|
| in | <i>*home_path</i> | the path to the home directory |
|----|-------------------|--------------------------------|

Returns

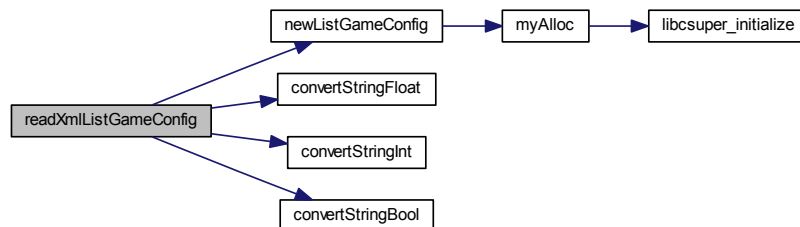
a [list_game_config](#)

Here is the call graph for this function:



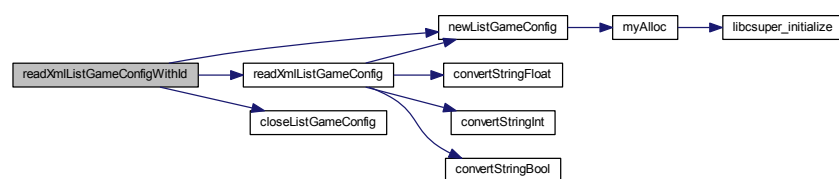
4.14.3.12 list_game_config* readXmlListGameConfig (char * filename)

Here is the call graph for this function:



4.14.3.13 list_game_config* readXmlListGameConfigWithId (char * filename, int * id, int nb_id)

Here is the call graph for this function:



4.14.3.14 `bool removeConfigFile (char * config_name, char * home_path)`

Depreciated, do nothing

Parameters

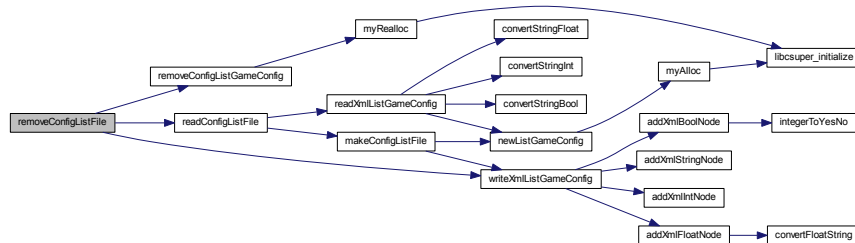
| | | |
|----|--------------------|--|
| in | <i>config_name</i> | the name of the game configuration which will be deleted |
| in | <i>home_path</i> | the path to the home directory |

Returns

true

4.14.3.15 `bool removeConfigListFile (int index_delete, list_game_config * ptr_list_config, char * home_path)`

Here is the call graph for this function:



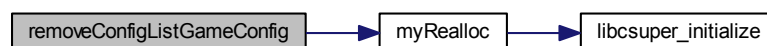
4.14.3.16 `void removeConfigListGameConfig (list_game_config * ptr_list_config, game_config config)`

Remove a config to the config list

Parameters

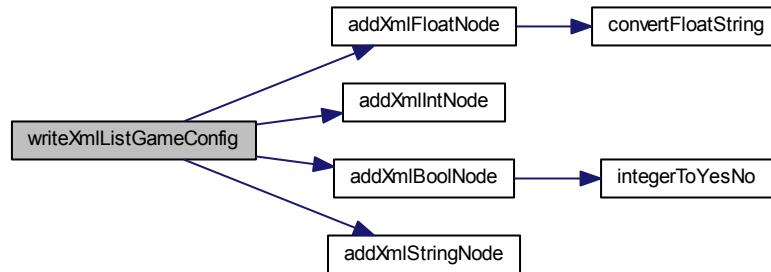
| | | |
|----|-------------------------|---|
| in | <i>*ptr_list_config</i> | a pointer on a list of game configuration |
| in | <i>config</i> | a game configuration |

Here is the call graph for this function:



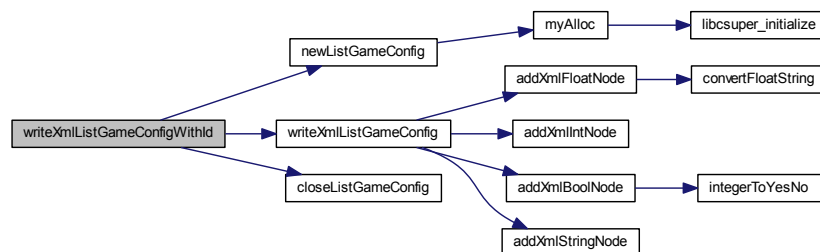
4.14.3.17 `bool writeXmlListGameConfig (char * filename, list_game_config * ptr_list_config)`

Here is the call graph for this function:



4.14.3.18 `bool writeXmlListGameConfigWithId (char * filename, list_game_config * ptr_list_config, int * id, int nb_id)`

Here is the call graph for this function:



4.15 gnuplot.c File Reference

```
#include "gnuplot.h"
```

Functions

- `bool exportToGnuplotFile (csuStruct *ptr_csu_struct, char *filename)`
- `bool exportToGnuplotData (csuStruct *ptr_csu_struct, char *filename)`
- `bool exportToGnuplotScript (csuStruct *ptr_csu_struct, char *filename)`

4.15.1 Detailed Description

Author

Remi BERTHO

Date

14/05/15

Version

4.2.0

4.15.2 Function Documentation

4.15.2.1 `bool exportToGnuplotData (csuStruct * ptr_csu_struct, char * filename)`

Export the csu structure to a gnuplot data file

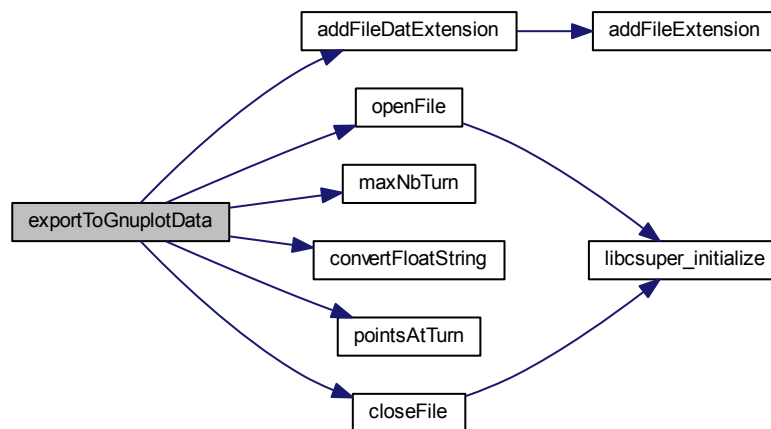
Parameters

| | | |
|----|-----------------------|--|
| in | <i>ptr_csu_struct</i> | a pointer on a csuStruct |
| in | <i>filename</i> | the filename of the gnuplot data file |

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:

4.15.2.2 `bool exportToGnuplotFile (csuStruct * ptr_csu_struct, char * filename)`

Export the csu structure to gnuplot files

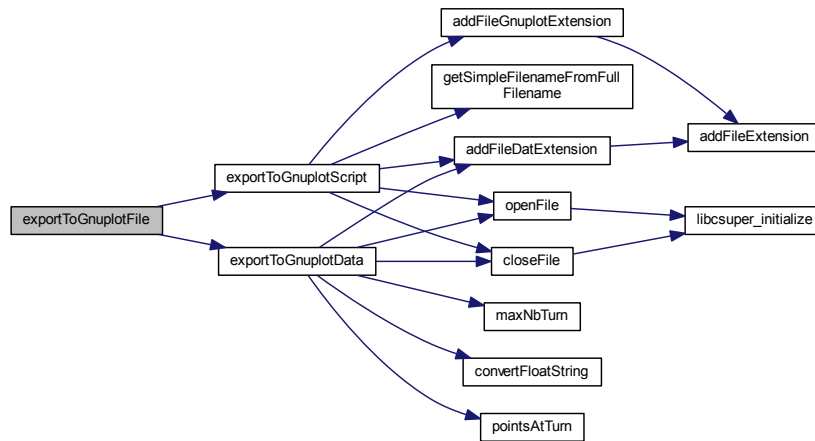
Parameters

| | | |
|----|-----------------------|--|
| in | <i>ptr_csu_struct</i> | a pointer on a csuStruct |
| in | <i>filename</i> | the filename |

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:



4.15.2.3 bool exportToGnuplotScript (csuStruct * ptr_csu_struct, char * filename)

Export the csu structure to a gnuplot script file

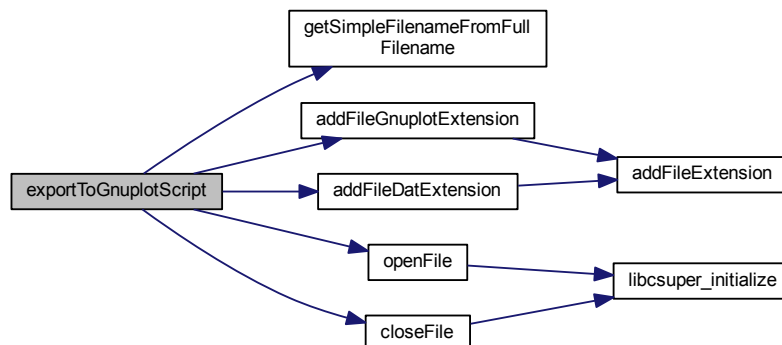
Parameters

| | | |
|----|-----------------------|--|
| in | <i>ptr_csu_struct</i> | a pointer on a csuStruct |
| in | <i>filename</i> | the filename of the gnuplot script file |

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:



4.16 gnuplot.h File Reference

```
#include "share.h"
#include "csu_struct.h"
#include "csu_files.h"
```

Functions

- bool [exportToGnuplotFile](#) ([csuStruct](#) *ptr_csu_struct, char *filename)
- bool [exportToGnuplotData](#) ([csuStruct](#) *ptr_csu_struct, char *filename)
- bool [exportToGnuplotScript](#) ([csuStruct](#) *ptr_csu_struct, char *filename)

4.16.1 Detailed Description

Author

Remi BERTHO

Date

14/05/15

Version

4.2.0

4.16.2 Function Documentation

4.16.2.1 bool [exportToGnuplotData](#) ([csuStruct](#) * *ptr_csu_struct*, char * *filename*)

Export the csu structure to a gnuplot data file

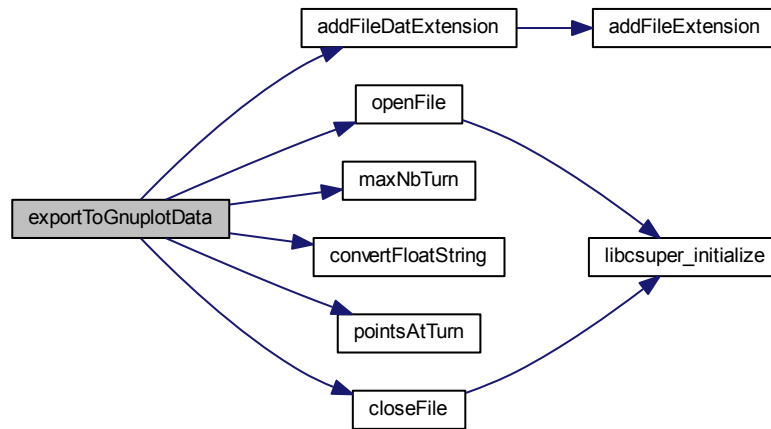
Parameters

| | | |
|----|-----------------------|--|
| in | <i>ptr_csu_struct</i> | a pointer on a csuStruct |
| in | <i>filename</i> | the filename of the gnuplot data file |

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:



4.16.2.2 bool exportToGnuplotFile (csuStruct * ptr_csu_struct, char * filename)

Export the csu structure to gnuplot files

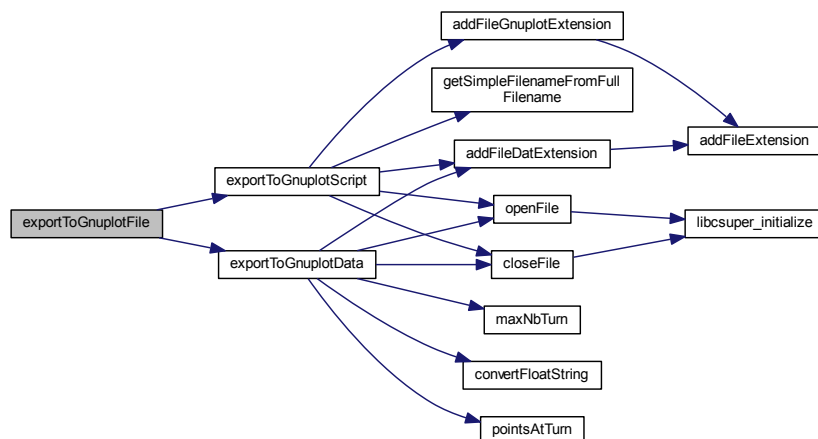
Parameters

| | | |
|----|-----------------------|--|
| in | <i>ptr_csu_struct</i> | a pointer on a csuStruct |
| in | <i>filename</i> | the filename |

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:



4.16.2.3 bool exportToGnuplotScript (csuStruct * ptr_csu_struct, char * filename)

Export the csu structure to a gnuplot script file

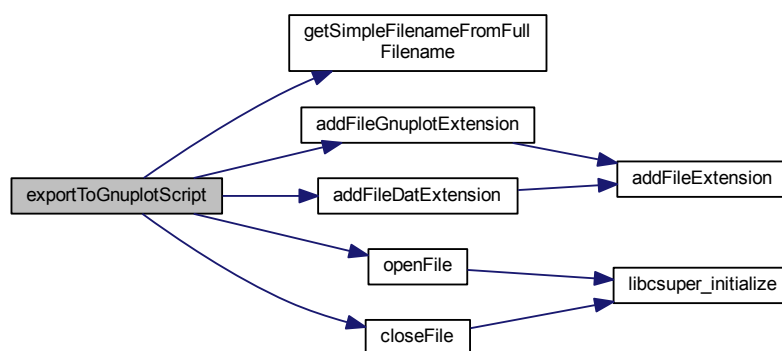
Parameters

| | | |
|----|-----------------------|--|
| in | <i>ptr_csu_struct</i> | a pointer on a csuStruct |
| in | <i>filename</i> | the filename of the gnuplot script file |

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:



4.17 libcsuper.h File Reference

Inclusion of all header files of libcsuper.

```

#include "csu_struct.h"
#include "share.h"
#include "csu_files.h"
#include "preferences_files.h"
#include "main_argument.h"
#include "game_config.h"
#include "file.h"
#include "filename.h"
#include "export.h"
#include "calculator.h"
#include "gnuplot.h"

```

Macros

- `#define` [NOT_LIBCSUPER](#)

4.17.1 Detailed Description

Inclusion of all header files of libcsuper.

Author

Remi BERTHO

Date

25/08/14

Version

4.0.2

4.17.2 Macro Definition Documentation

4.17.2.1 #define NOT_LIBCSUPER

Define that we don't compile libcsuper

4.18 main_argument.c File Reference

Begin csuper.

```
#include "main_argument.h"
```

Functions

- bool [searchArgument](#) (int argc, char *argv[], [main_argument_function](#) *function, int *file_place)
- void [displayHelp](#) ()

4.18.1 Detailed Description

Begin csuper.

Author

Remi BERTHO

Date

16/04/14

Version

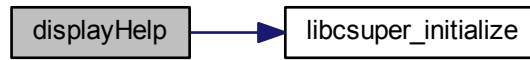
2.2.0

4.18.2 Function Documentation

4.18.2.1 void displayHelp ()

Display the help

Here is the call graph for this function:



4.18.2.2 `bool searchArgument (int argc, char * argv[], main_argument_function * function, int * file_place)`

Search the argument passed to the main function

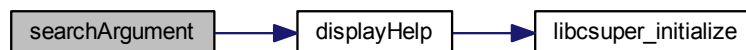
Parameters

| | | |
|----|-------------------|---|
| in | <i>argc</i> | the number of argument |
| in | <i>argv</i> | the array of argument |
| in | <i>function</i> | integer which determine which function run |
| in | <i>file_place</i> | integer which determine the index of the filename |

Returns

true if the function founded an argument, false otherwise

Here is the call graph for this function:



4.19 `main_argument.h` File Reference

Begin csuper.

```
#include "share.h"
```

Macros

- `#define STRING_READ_FILE "--read"`
- `#define STRING_READ_FILE_RED "-r"`
- `#define STRING_OPEN_FILE "--open"`
- `#define STRING_OPEN_FILE_RED "-o"`
- `#define STRING_HELP "--help"`
- `#define STRING_HELP_RED "-h"`
- `#define STRING_EXPORT_TO_PDF "--to-pdf"`

- `#define STRING_EXPORT_TO_PDF_RED "-p"`
- `#define STRING_EXPORT_TO_CSV "--to-csv"`
- `#define STRING_EXPORT_TO_CSV_RED "-c"`
- `#define STRING_EXPORT_TO_GNUPLOT "--to-gnuplot"`
- `#define STRING_EXPORT_TO_GNUPLOT_RED "-g"`
- `#define STRING_EXPORT_TO_M "--to-matlab"`
- `#define STRING_EXPORT_TO_M_RED "-m"`

Enumerations

- enum `main_argument_function` {
 `read_file`, `open_file`, `help`, `export_to_pdf`,
 `export_to_gnuplot`, `export_to_m`, `export_to_csv` }

Functions

- bool `searchArgument` (int argc, char *argv[], `main_argument_function` *function, int *file_place)
- void `displayHelp` ()

4.19.1 Detailed Description

Begin csuper.

Author

Remi BERTHO

Date

16/04/14

Version

2.2.0

4.19.2 Macro Definition Documentation

4.19.2.1 `#define STRING_EXPORT_TO_CSV "--to-csv"`

Define the argument which call to export to csv to "--to-csv"

4.19.2.2 `#define STRING_EXPORT_TO_CSV_RED "-c"`

Define the reduce argument which call to export to csv to "-c"

4.19.2.3 `#define STRING_EXPORT_TO_GNUPLOT "--to-gnuplot"`

Define the argument which call to export to gnuplot

4.19.2.4 `#define STRING_EXPORT_TO_GNUPLOT_RED "-g"`

Define the reduce argument which call to export to gnuplot

4.19.2.5 `#define STRING_EXPORT_TO_M "--to-matlab"`

Define the argument which call to export to matlab/octave file

4.19.2.6 `#define STRING_EXPORT_TO_M_RED "-m"`

Define the reduce argument which call to export to matlab/octave file

4.19.2.7 `#define STRING_EXPORT_TO_PDF "--to-pdf"`

Define the argument which call to export to pdf to "--to-pdf"

4.19.2.8 `#define STRING_EXPORT_TO_PDF_RED "-p"`

Define the reduce argument which call to export to pdf to "-p"

4.19.2.9 `#define STRING_HELP "--help"`

Define the argument which call help to "--help"

4.19.2.10 `#define STRING_HELP_RED "-h"`

Define the reduce argument which call help to "-h"

4.19.2.11 `#define STRING_OPEN_FILE "--open"`

Define the argument which call to open a file to "--open"

4.19.2.12 `#define STRING_OPEN_FILE_RED "-o"`

Define the reduce argument which call to open a file to "-o"

4.19.2.13 `#define STRING_READ_FILE "--read"`

Define the argument which call to read a file to "--read"

4.19.2.14 `#define STRING_READ_FILE_RED "-r"`

Define the reduce argument which call to read a file to "-r"

4.19.3 Enumeration Type Documentation

4.19.3.1 `enum main_argument_function`

Enumerator

read_file

open_file

help

export_to_pdf

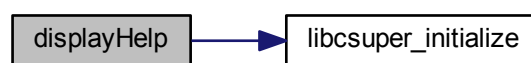
export_to_gnuplot
export_to_m
export_to_csv

4.19.4 Function Documentation

4.19.4.1 void displayHelp ()

Display the help

Here is the call graph for this function:



4.19.4.2 bool searchArgument (int argc, char * argv[], main_argument_function * function, int * file_place)

Search the argument passed to the main function

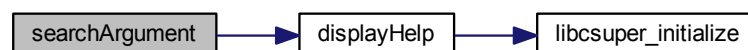
Parameters

| | | |
|----|-------------------|---|
| in | <i>argc</i> | the number of argument |
| in | <i>argv</i> | the array of argument |
| in | <i>function</i> | integer which determine which function run |
| in | <i>file_place</i> | integer which determine the index of the filename |

Returns

true if the function founded an argument, false otherwise

Here is the call graph for this function:



4.20 preferences_files.c File Reference

Function which store preferences into files.

```
#include "preferences_files.h"
#include <libxml/tree.h>
```

Functions

- void [createPreferencesFolder](#) (char *home_path)
- bool [createFileToolbarButtonPreferences](#) (char *home_path, [toolbar_button_preferences_struct](#) toolbar)
- bool [readFileToolbarButtonPreferences](#) (char *home_path, [toolbar_button_preferences_struct](#) *toolbar)
- bool [differeentsToolbarButtonPreferencesStruct](#) ([toolbar_button_preferences_struct](#) toolbar1, [toolbar_button_preferences_struct](#) toolbar2)
- bool [createFileMainWidowSize](#) (char *home_path, [main_window_size](#) size)
- bool [readFileMainWidowSize](#) (char *home_path, [main_window_size](#) *size)
- bool [createFileSystemPath](#) ()
- bool [readFileSystemPath](#) (char *file_name)
- bool [readSystemPath](#) (char *file_name)
- bool [changeSystemPath](#) (char *new_path)
- bool [createFileDifferenceBetweenPlayer](#) (char *home_path, [difference_between_player](#) diff)
- bool [readFileDifferenceBetweenPlayer](#) (char *home_path, [difference_between_player](#) *diff)
- bool [createFilePdfPreferences](#) (char *home_path, [export_pdf_preferences](#) *ptr_pref)
- bool [readFilePdfPreferences](#) (char *home_path, [export_pdf_preferences](#) *ptr_pref)
- bool [differeentsTExportPdfPreferencesStruct](#) ([export_pdf_preferences](#) pdf_1, [export_pdf_preferences](#) pdf_2)
- bool [createFileScoreDisplay](#) (char *home_path, [score_display](#) score)
- bool [readFileScoreDisplay](#) (char *home_path, [score_display](#) *score)
- bool [createFileMainWindowSide](#) (char *home_path, [main_window_side](#) pref)
- bool [readFileMainWindowSide](#) (char *home_path, [main_window_side](#) *pref)
- bool [createFileChartExportation](#) (char *home_path, [chart_exportation](#) pref)
- bool [readFileChartExportation](#) (char *home_path, [chart_exportation](#) *pref)
- bool [differeentsChartExportationStruct](#) ([chart_exportation](#) pref_1, [chart_exportation](#) pref_2)
- bool [writeXmlPreferencesFile](#) ([preferences](#) *pref, char *home_path)
- bool [writeXmlPreferencesFileType](#) ([one_preferences](#) *pref, char *home_path, [preferences_type](#) type)
- void [readXmlPreferencesFile](#) ([preferences](#) *pref, char *home_path)
- void [readXmlPreferencesFileType](#) ([one_preferences](#) *pref, char *home_path, [preferences_type](#) type)

4.20.1 Detailed Description

Function which store preferences into files.

Author

Remi BERTHO

Date

07/01/15

Version

4.2.0

4.20.2 Function Documentation

4.20.2.1 bool [changeSystemPath](#) (char * *new_path*)

Change the system path

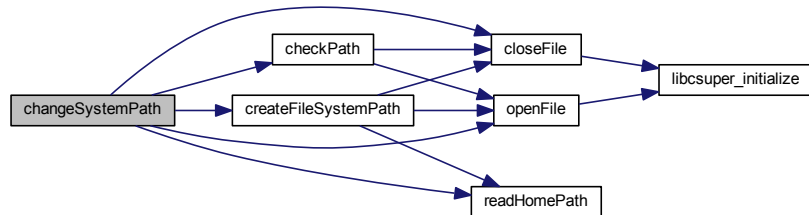
Parameters

| | | |
|---------|------------------|--------------|
| in, out | <i>*new_path</i> | the new path |
|---------|------------------|--------------|

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:



4.20.2.2 bool createFileChartExportation (char * home_path, chart_exportation pref)

Create the file which contain the chart exportation preferences

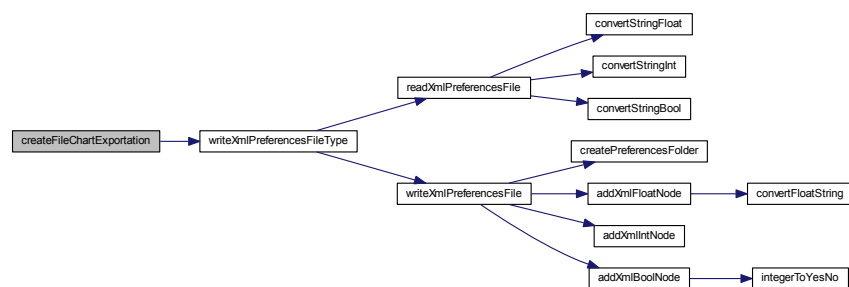
Parameters

| | | |
|----|------------------|---|
| in | <i>home_path</i> | the path to the home directory |
| in | <i>pref</i> | the chart_exportation structure |

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:



4.20.2.3 bool createFileDifferenceBetweenPlayer (char * home_path, difference_between_player diff)

Create the file which contain the data which explain that we display the differences between players

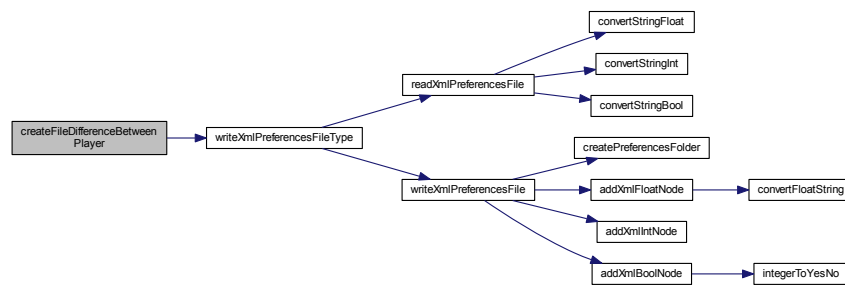
Parameters

| | | |
|----|------------------|--------------------------------|
| in | <i>home_path</i> | the path to the home directory |
| in | <i>diff</i> | the difference structure |

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:



4.20.2.4 bool createFileMainWidowSize (char * *home_path*, main_window_size size)

Create the file which contain the main window size

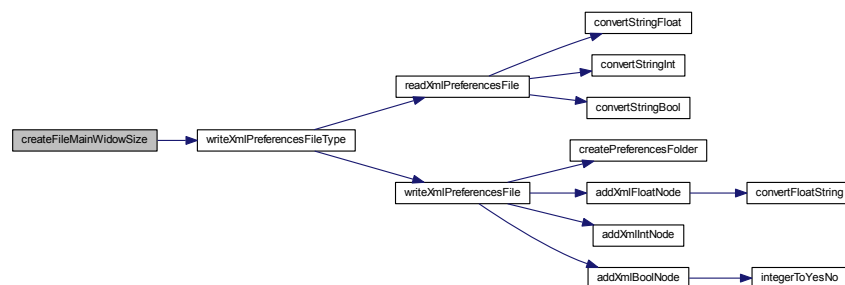
Parameters

| | | |
|----|------------------|--------------------------------|
| in | <i>home_path</i> | the path to the home directory |
| in | <i>size</i> | the size of the main window |

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:



4.20.2.5 bool createFileMainWindowSide (char * *home_path*, main_window_side pref)

Create the file which contain the data which explain what will be display in the left side of the main window

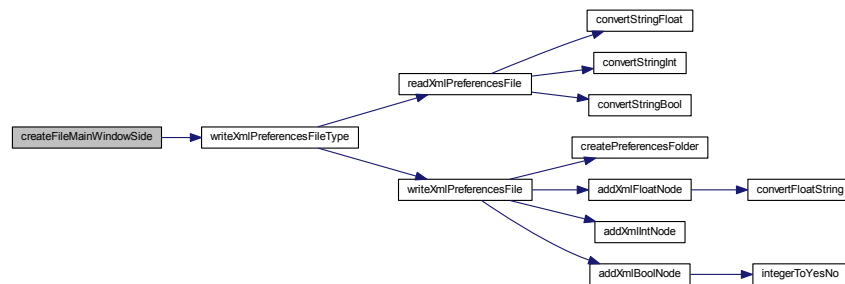
Parameters

| | | |
|----|------------------|--|
| in | <i>home_path</i> | the path to the home directory |
| in | <i>pref</i> | the main_window_side structure |

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:

4.20.2.6 bool createFilePdfPreferences (char * *home_path*, export_pdf_preferences * *ptr_pref*)

Create the file which contain the preferences export into a pdf file

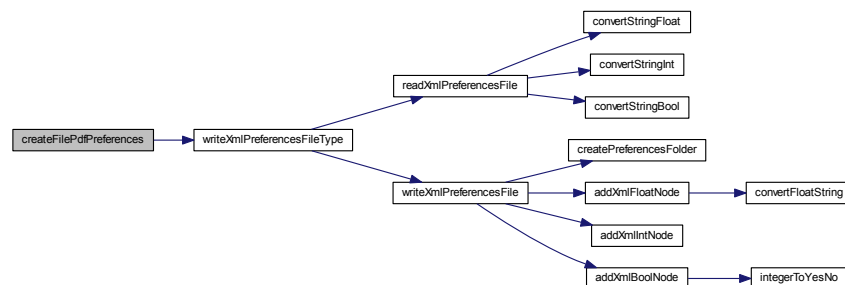
Parameters

| | | |
|----|------------------|---|
| in | <i>home_path</i> | the path to the home directory |
| in | <i>ptr_pref</i> | a pointer on a export_pdf_preferences |

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:

4.20.2.7 bool createFileScoreDisplay (char * *home_path*, score_display *score*)

Create the file which contain the data which explain that we display on the score grid

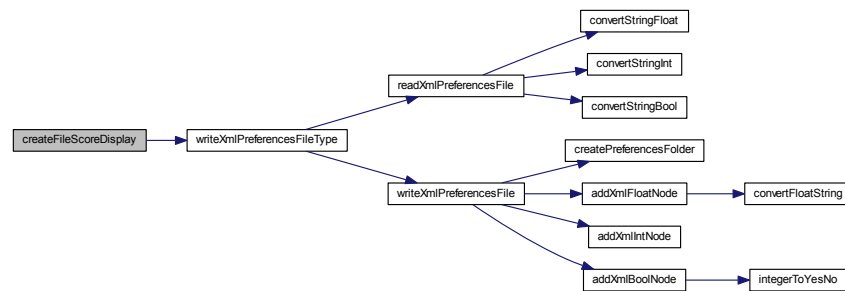
Parameters

| | | |
|----|------------------|---|
| in | <i>home_path</i> | the path to the home directory |
| in | <i>score</i> | the score_display structure |

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:



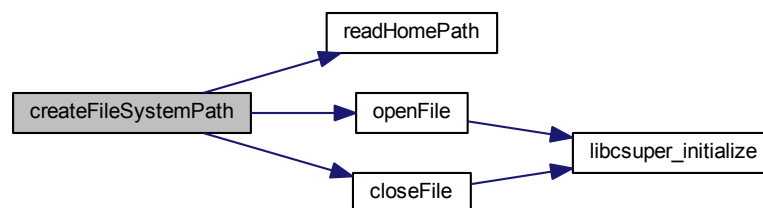
4.20.2.8 bool createFileSystemPath ()

Create the folder and the file which contain the system path

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:

4.20.2.9 bool createFileToolBarButtonPreferences (char * *home_path*, *toolbar_button_preferences_struct toolbar*)

Create the file which contain the preferences for the toolbar button

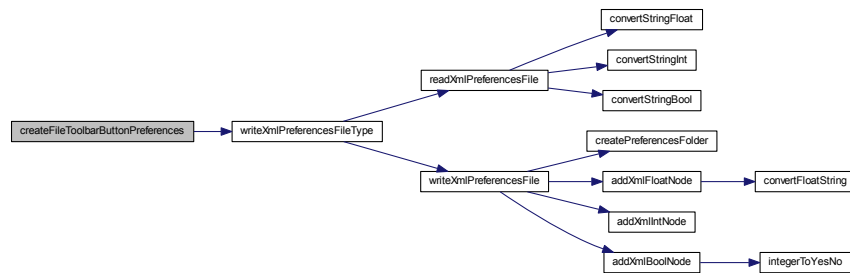
Parameters

| | | |
|----|------------------|---|
| in | <i>home_path</i> | the path to the home directory |
| in | <i>toolbar</i> | the toolbar button preferences for displaying |

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:

**4.20.2.10 void createPreferencesFolder (char * home_path)**

Create the folder which contain all preferences

Parameters

| | | |
|----|------------------|--------------------------------|
| in | <i>home_path</i> | the path to the home directory |
|----|------------------|--------------------------------|

4.20.2.11 bool differsChartExportationStruct (chart_exportation pref_1, chart_exportation pref_2)

Test if the two chart exportation structure

Parameters

| | | |
|----|---------------|--|
| in | <i>pref_1</i> | the first chart_exportation |
| in | <i>pref_2</i> | the second chart_exportation |

Returns

true if everything is OK, false otherwise

4.20.2.12 bool differsTExportPdfPreferencesStruct (export_pdf_preferences pdf_1, export_pdf_preferences pdf_2)

Test if the two pdf export preferences are different

Parameters

| | | |
|----|--------------|-----------------------------------|
| in | <i>pdf_1</i> | the first pdf export preferences |
| in | <i>pdf_2</i> | the second pdf export preferences |

Returns

true if everything is OK, false otherwise

4.20.2.13 `bool differsToolbarButtonPreferencesStruct (toolbar_button_preferences_struct toolbar1, toolbar_button_preferences_struct toolbar2)`

Test if the two toolbar button preferences are different

Parameters

| | | |
|----|-----------------|---------------------------------------|
| in | <i>toolbar1</i> | the first toolbar button preferences |
| in | <i>toolbar2</i> | the second toolbar button preferences |

Returns

true if everything is OK, false otherwise

4.20.2.14 `bool readFileChartExportation (char * home_path, chart_exportation * pref)`

Read the file which contain the chart exportation preferences

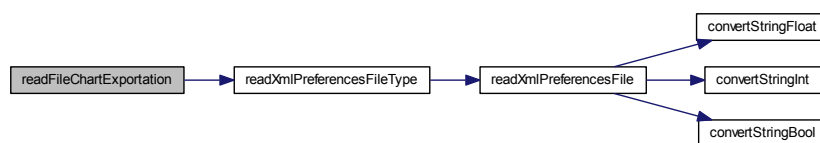
Parameters

| | | |
|----|------------------|---|
| in | <i>home_path</i> | the path to the home directory |
| in | <i>pref</i> | the chart_exportation structure |

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:



4.20.2.15 `bool readFileDifferenceBetweenPlayer (char * home_path, difference_between_player * diff)`

Read the file which contain the data which explain that we display the differences between players

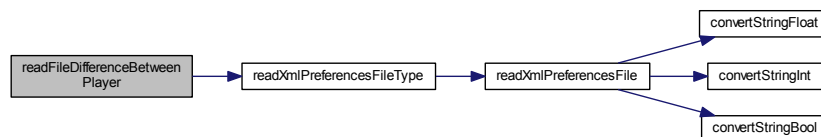
Parameters

| | | |
|----|------------------|--------------------------------|
| in | <i>home_path</i> | the path to the home directory |
| in | <i>diff</i> | the difference structure |

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:



4.20.2.16 bool readFileMainWidowSize (char * *home_path*, main_window_size * *size*)

Read the file which contain the main window size

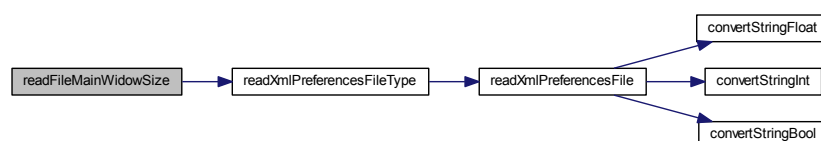
Parameters

| | | |
|----|------------------|--------------------------------|
| in | <i>home_path</i> | the path to the home directory |
| in | <i>size</i> | the size of the main window |

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:



4.20.2.17 bool readFileMainWindowSide (char * *home_path*, main_window_side * *pref*)

Read the file which contain the data which explain what will be display in the left side of the main window

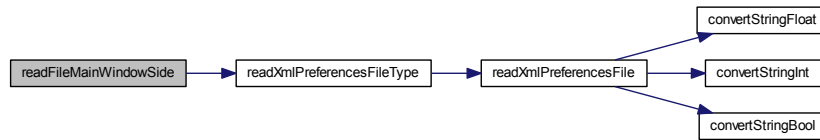
Parameters

| | | |
|----|------------------|--|
| in | <i>home_path</i> | the path to the home directory |
| in | <i>pref</i> | the main_window_side structure |

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:



4.20.2.18 bool readFilePdfPreferences (char * *home_path*, export_pdf_preferences * *ptr_pref*)

Read the file which contain the preferences export into a pdf file

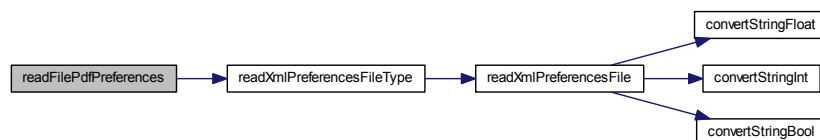
Parameters

| | | |
|----|------------------|---|
| in | <i>home_path</i> | the path to the home directory |
| in | <i>ptr_pref</i> | a pointer on a export_pdf_preferences |

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:



4.20.2.19 bool readFileScoreDisplay (char * *home_path*, score_display * *score*)

Read the file which contain the data which explain that we display on the score grid

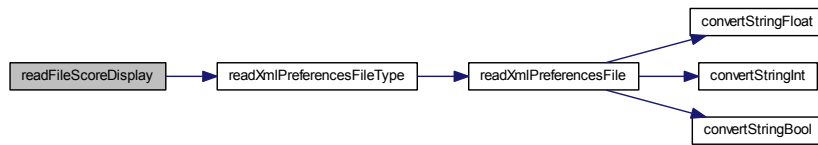
Parameters

| | | |
|----|------------------|---|
| in | <i>home_path</i> | the path to the home directory |
| in | <i>score</i> | the score_display structure |

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:



4.20.2.20 bool readFileSystemPath (char * file_name)

Read the system path and the path read to the filename

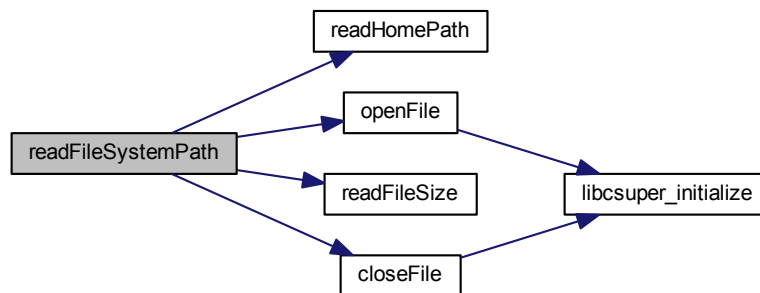
Parameters

| | | |
|---------|------------|--------------|
| in, out | *file_name | the filename |
|---------|------------|--------------|

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:



4.20.2.21 bool readFileToolBarButtonPreferences (char * home_path, toolbar_button_preferences_struct * toolbar)

Read the file which contain the preferences for the toolbar button

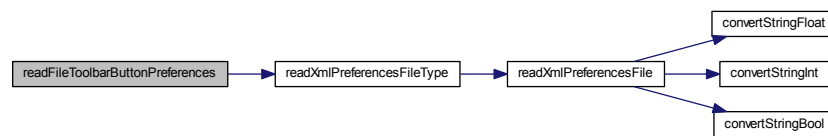
Parameters

| | | |
|----|------------------|---|
| in | <i>home_path</i> | the path to the home directory |
| in | <i>toolbar</i> | the toolbar button preferences for displaying |

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:

**4.20.2.22 bool readSystemPath (char * file_name)**

Add the system path, if the file system path doesn't exist, it create it.

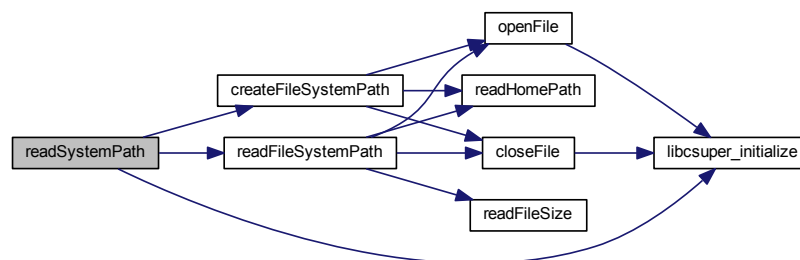
Parameters

| | | |
|---------|-------------------|--------------|
| in, out | <i>*file_name</i> | the filename |
|---------|-------------------|--------------|

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:

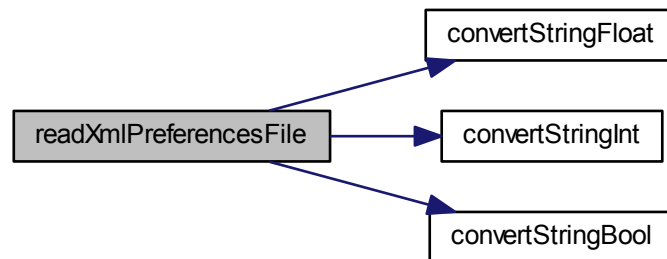
**4.20.2.23 bool readXmlPreferencesFile (preferences * pref, char * home_path)**

Read the preferences file

Parameters

| | | |
|----|------------------|--------------------------------|
| in | <i>home_path</i> | the path to the home directory |
| in | <i>pref</i> | the preferences structure |

Here is the call graph for this function:



4.20.2.24 `bool readXmlPreferencesFileType (one_preferences * pref, char * home_path, preferences_type type)`

Read the preferences file with the preferences selected

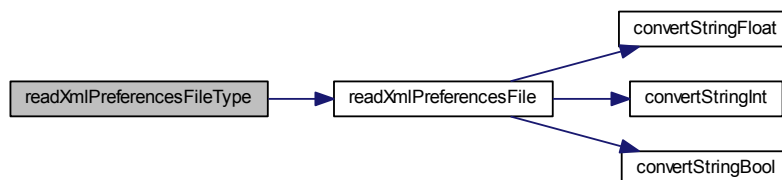
Parameters

| | | |
|----|------------------|--------------------------------|
| in | <i>home_path</i> | the path to the home directory |
| in | <i>pref</i> | a preferences |
| in | <i>type</i> | the type of preferences |

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:



4.20.2.25 `bool writeXmlPreferencesFile (preferences * pref, char * home_path)`

Write the preferences file

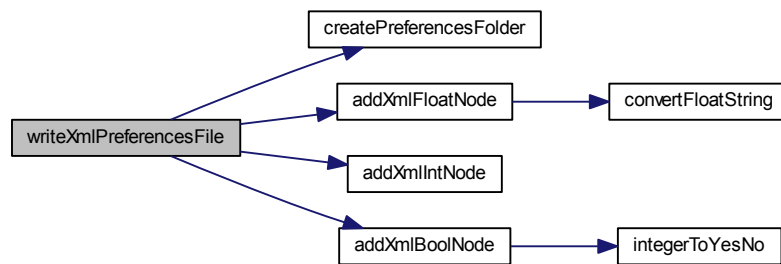
Parameters

| | | |
|----|------------------|--------------------------------|
| in | <i>home_path</i> | the path to the home directory |
| in | <i>pref</i> | the preferences structure |

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:



4.20.2.26 bool writeXmlPreferencesFileType (one_preferences * pref, char * home_path, preferences_type type)

Write the preferences file with the preferences selected

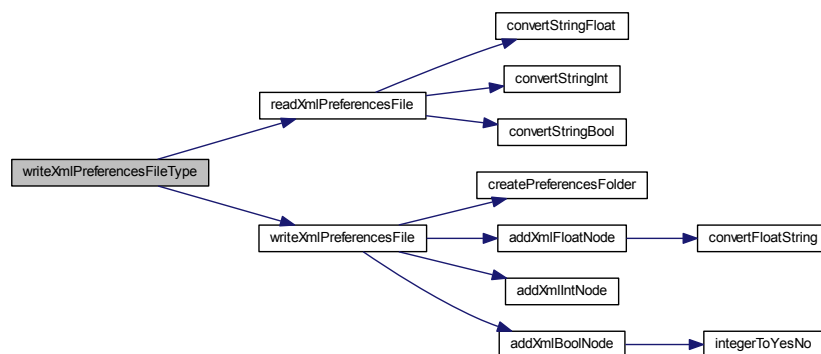
Parameters

| | | |
|----|------------------|--------------------------------|
| in | <i>home_path</i> | the path to the home directory |
| in | <i>pref</i> | a preferences |
| in | <i>type</i> | the type of preferences |

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:



4.21 preferences_files.h File Reference

Prototypes des fonctions qui l'emrankment des fichiers sauvegardes.

```
#include "share.h"
#include "csu_struct.h"
#include "csu_files.h"
#include "filename.h"
#include "export.h"
```

Data Structures

- struct [toolbar_button_preferences_struct](#)
- struct [main_window_size](#)
- struct [difference_between_player](#)
- struct [score_display](#)
- struct [main_window_side](#)
- struct [chart_exportation](#)
- struct [preferences](#)
- union [one_preferences](#)

Macros

- #define [FILENAME_SYSTEM_PATH](#) "system_path.txt"
- #define [FILENAME_TOOLBAR_BUTTON_PREFERENCES](#) "toolbar_button_preferences.txt"
- #define [FILENAME_MAIN_WINDOW_SIZE](#) "main_window_size.txt"
- #define [FILENAME_DIFFERENCE_BETWEEN_PLAYER](#) "difference_betetween_player.txt"
- #define [FILENAME_PDF_PREFERENCES](#) "pdf_preferences.txt"
- #define [FILENAME_SCORE_DISPLAY](#) "score_display_preferences.txt"
- #define [FILENAME_MAIN_WINDOW_SIDE](#) "main_window_side_preferences.txt"
- #define [PREFERENCES_FOLDER_NAME](#) ".csuper"
- #define [FILENAME_PREFERENCES_XML](#) "preferences.xml"
- #define [PREFERENCES_FILE_XML_VERSION](#) 1.0

Enumerations

- enum [preferences_type](#) {
[toolbar_type](#), [size_type](#), [diff_type](#), [score_type](#),
[side_type](#), [pdf_type](#), [chart_type](#) }

Functions

- void [createPreferencesFolder](#) (char *home_path)
- bool [createFileToolbarButtonPreferences](#) (char *home_path, [toolbar_button_preferences_struct](#) toolbar)
- bool [readFileToolbarButtonPreferences](#) (char *home_path, [toolbar_button_preferences_struct](#) *toolbar)
- bool [differeentsToolbarButtonPreferencesStruct](#) ([toolbar_button_preferences_struct](#) toolbar1, [toolbar_button_preferences_struct](#) toolbar2)
- bool [createFileMainWidowSize](#) (char *home_path, [main_window_size](#) size)
- bool [readFileMainWidowSize](#) (char *home_path, [main_window_size](#) *size)
- bool [createFileSystemPath](#) ()
- bool [readFileSystemPath](#) (char *file_name)
- bool [readSystemPath](#) (char *file_name)

- bool [changeSystemPath](#) (char *new_path)
- bool [createFileDifferenceBetweenPlayer](#) (char *home_path, [difference_between_player](#) diff)
- bool [readFileDifferenceBetweenPlayer](#) (char *home_path, [difference_between_player](#) *diff)
- bool [createFilePdfPreferences](#) (char *home_path, [export_pdf_preferences](#) *ptr_pref)
- bool [readFilePdfPreferences](#) (char *home_path, [export_pdf_preferences](#) *ptr_pref)
- bool [differentsTExportPdfPreferencesStruct](#) ([export_pdf_preferences](#) pdf_1, [export_pdf_preferences](#) pdf_2)
- bool [createFileScoreDisplay](#) (char *home_path, [score_display](#) score)
- bool [readFileScoreDisplay](#) (char *home_path, [score_display](#) *score)
- bool [createFileMainWindowSide](#) (char *home_path, [main_window_side](#) pref)
- bool [readFileMainWindowSide](#) (char *home_path, [main_window_side](#) *pref)
- bool [createFileChartExportation](#) (char *home_path, [chart_exportation](#) pref)
- bool [readFileChartExportation](#) (char *home_path, [chart_exportation](#) *pref)
- bool [differentsChartExportationStruct](#) ([chart_exportation](#) pref_1, [chart_exportation](#) pref_2)
- bool [writeXmlPreferencesFile](#) ([preferences](#) *pref, char *home_path)
- bool [writeXmlPreferencesFileType](#) ([one_preferences](#) *pref, char *home_path, [preferences_type](#) type)
- void [readXmlPreferencesFile](#) ([preferences](#) *pref, char *home_path)
- void [readXmlPreferencesFileType](#) ([one_preferences](#) *pref, char *home_path, [preferences_type](#) type)

4.21.1 Detailed Description

Prototypes des fonctions qui l'emrankment des fichiers sauvegardes.

Author

Remi BERTHO

Date

07/01/15

Version

4.2.0

4.21.2 Macro Definition Documentation

4.21.2.1 `#define FILENAME_DIFFERENCE_BETWEEN_PLAYER "difference_between_player.txt"`

Define the filename of the file which contain the difference preference

4.21.2.2 `#define FILENAME_MAIN_WINDOW_SIDE "main_window_side_preferences.txt"`

Define the filename of the file which contain the main window side preferences

4.21.2.3 `#define FILENAME_MAIN_WINDOW_SIZE "main_window_size.txt"`

4.21.2.4 `#define FILENAME_PDF_PREFERENCES "pdf_preferences.txt"`

Define the filename of the file which contain pdf preferences

4.21.2.5 `#define FILENAME_PREFERENCES_XML "preferences.xml"`

Define the name of the xml preferences file

4.21.2.6 `#define FILENAME_SCORE_DISPLAY "score_display_preferences.txt"`

Define the filename of the file which contain score display preferences

4.21.2.7 `#define FILENAME_SYSTEM_PATH "system_path.txt"`

Define the filename of the file which contain the system path

4.21.2.8 `#define FILENAME_TOOLBAR_BUTTON_PREFERENCES "toolbar_button_preferences.txt"`

Define the filename of the file which contain the toolbar button preferences

4.21.2.9 `#define PREFERENCES_FILE_XML_VERSION 1.0`

Define the version of the XML preferences file

4.21.2.10 `#define PREFERENCES_FOLDER_NAME ".csuper"`

Define the folder name of the csuper preferences

4.21.3 Enumeration Type Documentation

4.21.3.1 `enum preferences_type`

Enumerator

toolbar_type

size_type

diff_type

score_type

side_type

pdf_type

chart_type

4.21.4 Function Documentation

4.21.4.1 `bool changeSystemPath (char * new_path)`

Change the system path

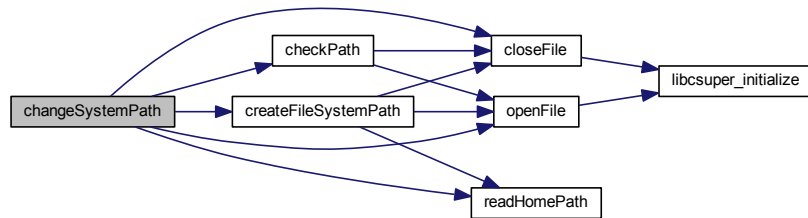
Parameters

| | | |
|----------------------|------------------------|--------------|
| <code>in, out</code> | <code>*new_path</code> | the new path |
|----------------------|------------------------|--------------|

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:



4.21.4.2 bool createFileChartExportation (char * *home_path*, chart_exportation *pref*)

Create the file which contain the chart exportation preferences

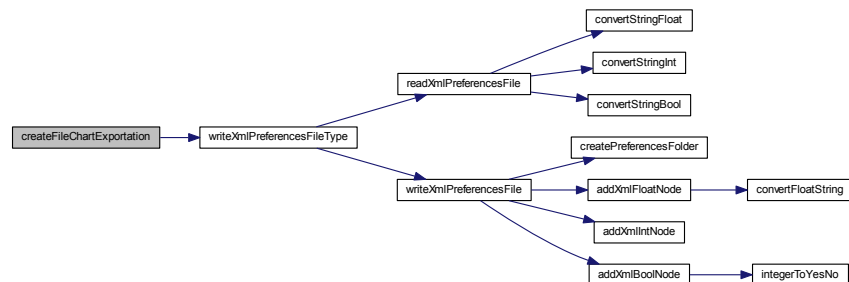
Parameters

| | | |
|----|------------------|---|
| in | <i>home_path</i> | the path to the home directory |
| in | <i>pref</i> | the chart_exportation structure |

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:



4.21.4.3 bool createFileDifferenceBetweenPlayer (char * *home_path*, difference_between_player *diff*)

Create the file which contain the data which explain that we display the differences between players

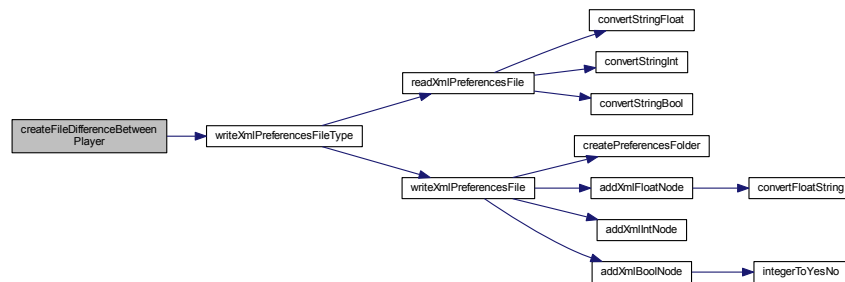
Parameters

| | | |
|----|------------------|--------------------------------|
| in | <i>home_path</i> | the path to the home directory |
| in | <i>diff</i> | the difference structure |

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:



4.21.4.4 bool createFileMainWidowSize (char * *home_path*, main_window_size *size*)

Create the file which contain the main window size

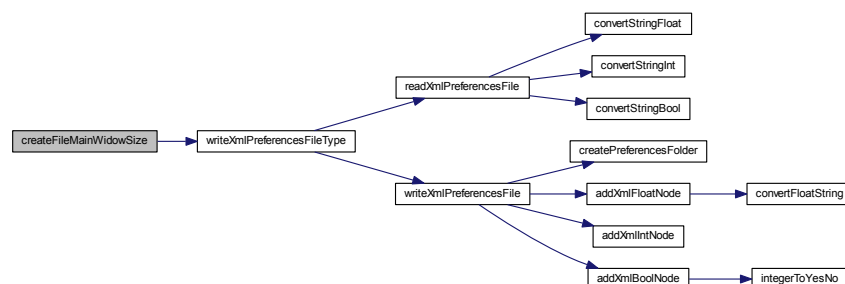
Parameters

| | | |
|----|------------------|--------------------------------|
| in | <i>home_path</i> | the path to the home directory |
| in | <i>size</i> | the size of the main window |

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:



4.21.4.5 bool createFileMainWindowSide (char * *home_path*, main_window_side *pref*)

Create the file which contain the data which explain what will be display in the left side of the main window

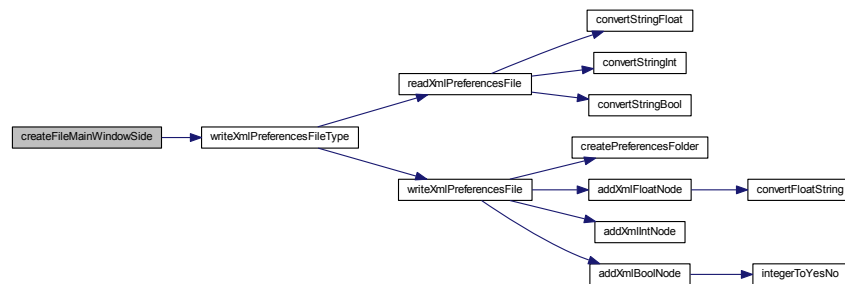
Parameters

| | | |
|----|------------------|--|
| in | <i>home_path</i> | the path to the home directory |
| in | <i>pref</i> | the main_window_side structure |

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:



4.21.4.6 bool createFilePdfPreferences (char * *home_path*, export_pdf_preferences * *ptr_pref*)

Create the file which contain the preferences export into a pdf file

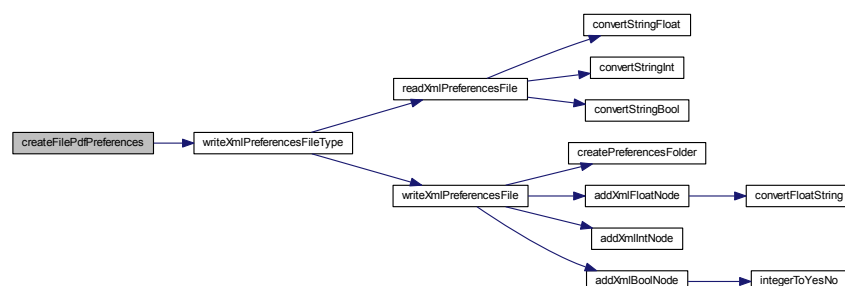
Parameters

| | | |
|----|------------------|---|
| in | <i>home_path</i> | the path to the home directory |
| in | <i>ptr_pref</i> | a pointer on a export_pdf_preferences |

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:



4.21.4.7 bool createFileScoreDisplay (char * *home_path*, score_display *score*)

Create the file which contain the data which explain that we display on the score grid

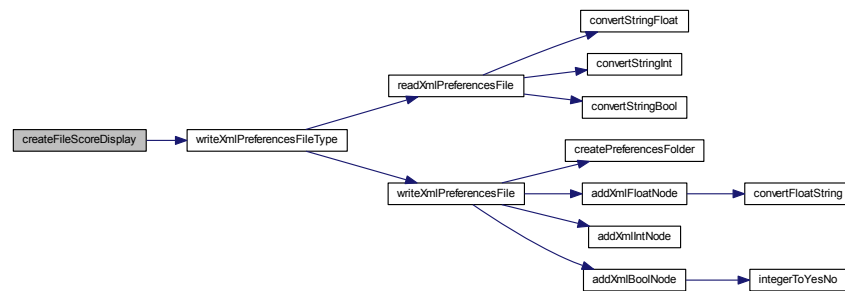
Parameters

| | | |
|----|------------------|---|
| in | <i>home_path</i> | the path to the home directory |
| in | <i>score</i> | the score_display structure |

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:



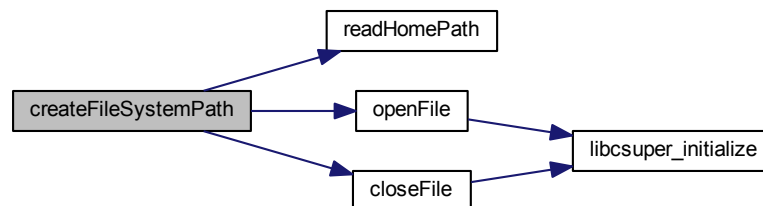
4.21.4.8 bool createFileSystemPath ()

Create the folder and the file which contain the system path

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:

4.21.4.9 bool createFileToolBarButtonPreferences (char * *home_path*, *toolbar_button_preferences_struct toolbar*)

Create the file which contain the preferences for the toolbar button

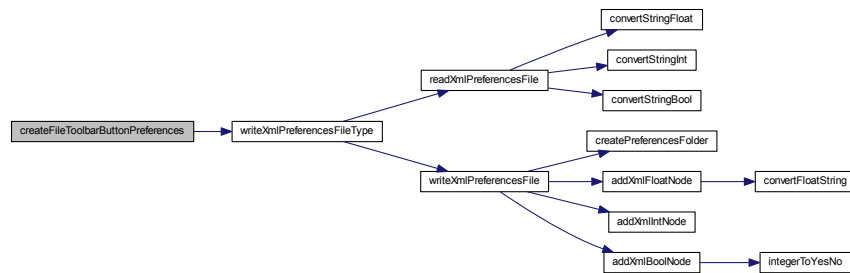
Parameters

| | | |
|----|------------------|---|
| in | <i>home_path</i> | the path to the home directory |
| in | <i>toolbar</i> | the toolbar button preferences for displaying |

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:



4.21.4.10 void createPreferencesFolder (char * *home_path*)

Create the folder which contain all preferences

Parameters

| | | |
|----|------------------|--------------------------------|
| in | <i>home_path</i> | the path to the home directory |
|----|------------------|--------------------------------|

4.21.4.11 bool differsChartExportationStruct (chart_exportation *pref_1*, chart_exportation *pref_2*)

Test if the two chart exportation structure

Parameters

| | | |
|----|---------------|--|
| in | <i>pref_1</i> | the first chart_exportation |
| in | <i>pref_2</i> | the second chart_exportation |

Returns

true if everything is OK, false otherwise

4.21.4.12 bool differsTExportPdfPreferencesStruct (export_pdf_preferences *pdf_1*, export_pdf_preferences *pdf_2*)

Test if the two pdf export preferences are different

Parameters

| | | |
|----|--------------|-----------------------------------|
| in | <i>pdf_1</i> | the first pdf export preferences |
| in | <i>pdf_2</i> | the second pdf export preferences |

Returns

true if everything is OK, false otherwise

4.21.4.13 `bool differsToolbarButtonPreferencesStruct (toolbar_button_preferences_struct toolbar1, toolbar_button_preferences_struct toolbar2)`

Test if the two toolbar button preferences are different

Parameters

| | | |
|----|-----------------|---------------------------------------|
| in | <i>toolbar1</i> | the first toolbar button preferences |
| in | <i>toolbar2</i> | the second toolbar button preferences |

Returns

true if everything is OK, false otherwise

4.21.4.14 `bool readFileChartExportation (char * home_path, chart_exportation * pref)`

Read the file which contain the chart exportation preferences

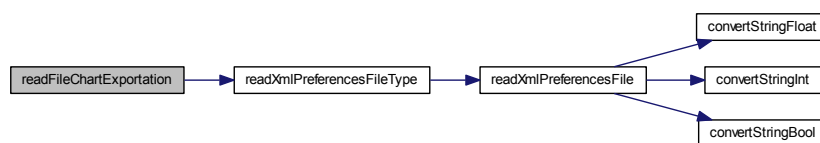
Parameters

| | | |
|----|------------------|---|
| in | <i>home_path</i> | the path to the home directory |
| in | <i>pref</i> | the chart_exportation structure |

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:



4.21.4.15 `bool readFileDifferenceBetweenPlayer (char * home_path, difference_between_player * diff)`

Read the file which contain the data which explain that we display the differences between players

Parameters

| | | |
|----|------------------|--------------------------------|
| in | <i>home_path</i> | the path to the home directory |
| in | <i>diff</i> | the difference structure |

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:



4.21.4.16 bool readFileMainWidowSize (char * *home_path*, main_window_size * *size*)

Read the file which contain the main window size

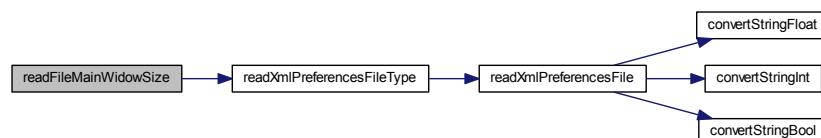
Parameters

| | | |
|----|------------------|--------------------------------|
| in | <i>home_path</i> | the path to the home directory |
| in | <i>size</i> | the size of the main window |

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:



4.21.4.17 bool readFileMainWindowSide (char * *home_path*, main_window_side * *pref*)

Read the file which contain the data which explain what will be display in the left side of the main window

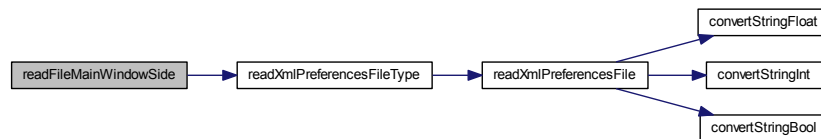
Parameters

| | | |
|----|------------------|--|
| in | <i>home_path</i> | the path to the home directory |
| in | <i>pref</i> | the main_window_side structure |

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:



4.21.4.18 bool readFilePdfPreferences (char * *home_path*, export_pdf_preferences * *ptr_pref*)

Read the file which contain the preferences export into a pdf file

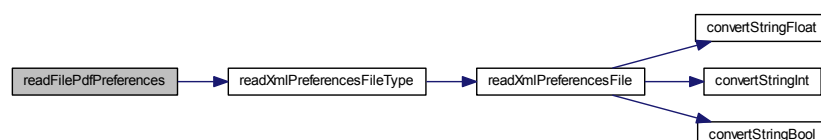
Parameters

| | | |
|----|------------------|---|
| in | <i>home_path</i> | the path to the home directory |
| in | <i>ptr_pref</i> | a pointer on a export_pdf_preferences |

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:



4.21.4.19 bool readFileScoreDisplay (char * *home_path*, score_display * *score*)

Read the file which contain the data which explain that we display on the score grid

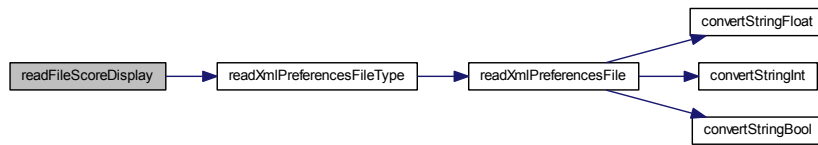
Parameters

| | | |
|----|------------------|---|
| in | <i>home_path</i> | the path to the home directory |
| in | <i>score</i> | the score_display structure |

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:



4.21.4.20 bool readFileSystemPath (char * *file_name*)

Read the system path and the path read to the filename

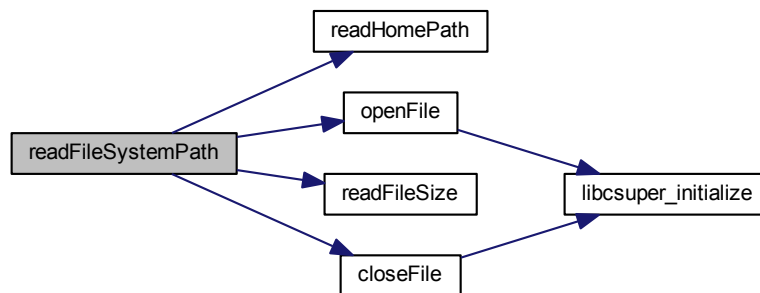
Parameters

| | | |
|---------|--------------------|--------------|
| in, out | * <i>file_name</i> | the filename |
|---------|--------------------|--------------|

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:



4.21.4.21 bool readFileToolBarButtonPreferences (char * *home_path*, toolbar_button_preferences_struct * *toolbar*)

Read the file which contain the preferences for the toolbar button

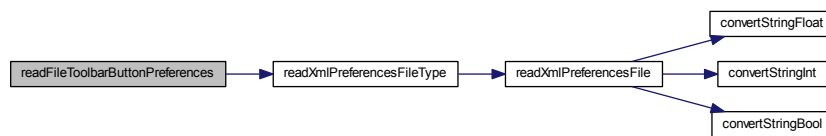
Parameters

| | | |
|----|------------------|---|
| in | <i>home_path</i> | the path to the home directory |
| in | <i>toolbar</i> | the toolbar button preferences for displaying |

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:

**4.21.4.22 bool readSystemPath (char * file_name)**

Add the system path, if the file system path doesn't exist, it create it.

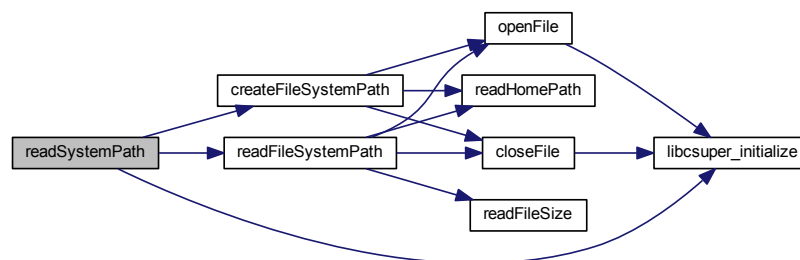
Parameters

| | | |
|---------|-------------------|--------------|
| in, out | <i>*file_name</i> | the filename |
|---------|-------------------|--------------|

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:

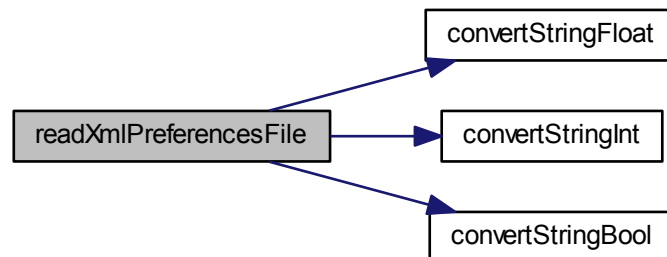
**4.21.4.23 void readXmlPreferencesFile (preferences * pref, char * home_path)**

Read the preferences file

Parameters

| | | |
|----|------------------|--------------------------------|
| in | <i>home_path</i> | the path to the home directory |
| in | <i>pref</i> | the preferences structure |

Here is the call graph for this function:



4.21.4.24 `void readXmlPreferencesFileType (one_preferences * pref, char * home_path, preferences_type type)`

Read the preferences file with the preferences selected

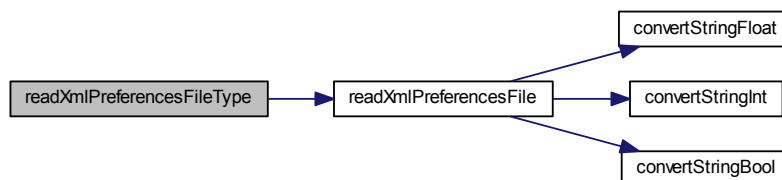
Parameters

| | | |
|----|------------------|--------------------------------|
| in | <i>home_path</i> | the path to the home directory |
| in | <i>pref</i> | a preferences |
| in | <i>type</i> | the type of preferences |

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:



4.21.4.25 `bool writeXmlPreferencesFile (preferences * pref, char * home_path)`

Write the preferences file

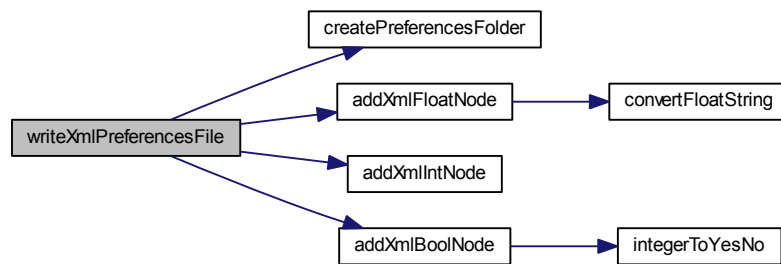
Parameters

| | | |
|----|------------------|--------------------------------|
| in | <i>home_path</i> | the path to the home directory |
| in | <i>pref</i> | the preferences structure |

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:



4.21.4.26 bool writeXmlPreferencesFileType (one_preferences * pref, char * home_path, preferences_type type)

Write the preferences file with the preferences selected

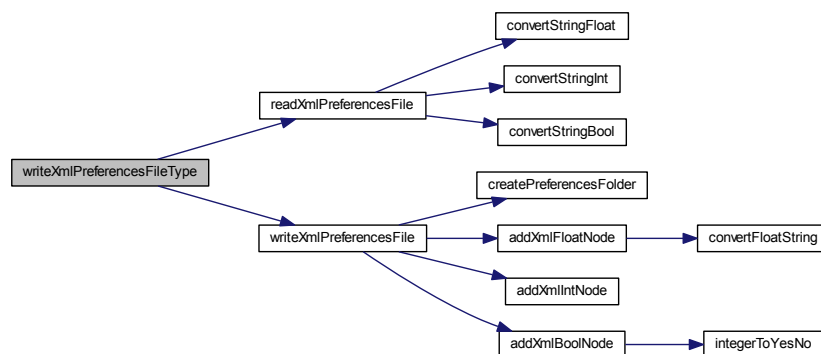
Parameters

| | | |
|----|------------------|--------------------------------|
| in | <i>home_path</i> | the path to the home directory |
| in | <i>pref</i> | a preferences |
| in | <i>type</i> | the type of preferences |

Returns

true if everything is OK, false otherwise

Here is the call graph for this function:



4.22 share.c File Reference

Essential function of libcsuper.

```
#include "share.h"
#include "csu_files.h"
```

Functions

- void [libcsuper_initialize](#) ()
- void [wrongChoice](#) ()
- void [clearScreen](#) ()
- int [compareFloatAscending](#) (void const *a, void const *b)
- int [compareFloatDescending](#) (void const *a, void const *b)
- void * [myAlloc](#) (int size_alloue)
- void [myRealloc](#) (void **ptr, int size_alloue)
- char * [integerToYesNo](#) (int i, char *yes, char *no)
- char * [utf8ToLatin9](#) (const char *const string)
- void [convertFloatString](#) (char *output, float input, int decimal_place)
- float [convertStringFloat](#) (char *str)
- int [convertStringInt](#) (char *str)
- bool [convertStringBool](#) (char *str)

4.22.1 Detailed Description

Essential function of libcsuper.

Author

Remi BERTHO

Date

25/01/15

Version

4.1.0

4.22.2 Function Documentation

4.22.2.1 void clearScreen ()

Clear the terminal.

Here is the call graph for this function:



4.22.2.2 int compareFloatAscending (void const * *a*, void const * *b*)

Compare 2 float

Parameters

| | | |
|----|-----------|----------------------|
| in | <i>*a</i> | a pointer on a float |
| in | <i>*b</i> | a pointer on a float |

Returns

1 if $a > b$, 0 if $a = b$ and -1 if $a < b$

4.22.2.3 int compareFloatDescending (void const * *a*, void const * *b*)

Compare 2 float

Parameters

| | | |
|----|-----------|----------------------|
| in | <i>*a</i> | a pointer on a float |
| in | <i>*b</i> | a pointer on a float |

Returns

1 if $a < b$, 0 if $a = b$ and -1 if $a > b$

4.22.2.4 void convertFloatString (char * *output*, float *input*, int *decimal_place*)

Convert a float into the output string with a specific number of decimal place

Parameters

| | | |
|----|----------------------|--|
| in | <i>output</i> | the output sting |
| in | <i>input</i> | the input float |
| in | <i>decimal_place</i> | the number of decimal, must be between 0 and 3 |

Returns

the ranking

4.22.2.5 bool convertStringBool (char * *str*)

Convert a string into a bool

Parameters

| | | |
|----|------------|------------|
| in | <i>str</i> | the string |
|----|------------|------------|

Returns

the bool

4.22.2.6 float convertStringFloat (char * *str*)

Convert a string into a float

Parameters

| | | |
|-----------|------------|------------|
| <i>in</i> | <i>str</i> | the string |
|-----------|------------|------------|

Returns

the float

4.22.2.7 int convertStringInt (char * *str*)

Convert a string into a int

Parameters

| | | |
|-----------|------------|------------|
| <i>in</i> | <i>str</i> | the string |
|-----------|------------|------------|

Returns

the int

4.22.2.8 char * integerToYesNo (int *i*, char * *yes*, char * *no*)

Transform an integer to yes or no

Parameters

| | | |
|-----------|------------|----------------|
| <i>in</i> | <i>i</i> | the integer |
| <i>in</i> | <i>yes</i> | the yes string |
| <i>in</i> | <i>no</i> | the no string |

Returns

yes if $i > 0$, no otherwise

4.22.2.9 void libcsuper_initialize ()

Initialize libcsuper with gettext.

4.22.2.10 void * myAlloc (int *size_alloue*)

Allocate a memory block and check if everything is OK.

Parameters

| | | |
|-----------|--------------------|----------|
| <i>in</i> | <i>size_alloue</i> | the size |
|-----------|--------------------|----------|

Returns

a pointer on the allocate memory block

Here is the call graph for this function:

**4.22.2.11 void myRealloc (void ** *ptr*, int *size_alloue*)**

Here is the call graph for this function:

**4.22.2.12 char * utf8ToLatin9 (const char *const *string*)**

Create a dynamically allocated copy of string, changing the encoding from UTF-8 to ISO-8859-15. Unsupported code points are ignored.

Parameters

| | | |
|-----------|---------------|---------------------------|
| <i>in</i> | <i>string</i> | the input string in UTF-8 |
|-----------|---------------|---------------------------|

Returns

a newly allocated string in ISO-8859-15

4.22.2.13 void wrongChoice ()

Display an error message.

Here is the call graph for this function:



4.23 share.h File Reference

Header for the essential function of libcsuper.

```

#include <sys/stat.h>
#include <sys/types.h>
#include <stdio.h>
#include <stdlib.h>
#include <errno.h>
#include <string.h>
#include <libintl.h>
#include <stdbool.h>
#include <hpdf.h>
#include <math.h>
#include <locale.h>

```

Macros

- `#define _(String) dgettext ("libcsuper", String)`
- `#define CSUPER_VERSION "4.2.0"`

Functions

- void `libcsuper_initialize` ()
- void `wrongChoice` ()
- void `clearScreen` ()
- int `compareFloatDescending` (void const *a, void const *b)
- int `compareFloatAscending` (void const *a, void const *b)
- void * `myAlloc` (int size_alloue)
- void `myRealloc` (void **ptr, int size_alloue)
- char * `integerToYesNo` (int i, char *yes, char *no)
- char * `utf8ToLatin9` (const char *const string)
- void `convertFloatString` (char *output, float input, int decimal_place)
- float `convertStringFloat` (char *str)
- int `convertStringInt` (char *str)
- bool `convertStringBool` (char *str)

4.23.1 Detailed Description

Header for the essential function of libcsuper.

Author

Remi BERTHO

Date

25/01/15

Version

4.1.0

4.23.2 Macro Definition Documentation**4.23.2.1 #define _(*String*) dgettext ("libcsuper", *String*)**

Define the _ for gettext.

4.23.2.2 #define CSUPER_VERSION "4.2.0"

Define the version of csuper

4.23.3 Function Documentation**4.23.3.1 void clearScreen ()**

Clear the terminal.

Here is the call graph for this function:

**4.23.3.2 int compareFloatAscending (void const * *a*, void const * *b*)**

Compare 2 float

Parameters

| | | |
|----|-----------|----------------------|
| in | <i>*a</i> | a pointer on a float |
| in | <i>*b</i> | a pointer on a float |

Returns

1 if $a > b$, 0 if $a = b$ and -1 if $a < b$

4.23.3.3 int compareFloatDescending (void const * *a*, void const * *b*)

Compare 2 float

Parameters

| | | |
|-----------|-----------|----------------------|
| <i>in</i> | <i>*a</i> | a pointer on a float |
| <i>in</i> | <i>*b</i> | a pointer on a float |

Returns

1 if $a < b$, 0 if $a = b$ and -1 if $a > b$

4.23.3.4 void convertFloatString (char * *output*, float *input*, int *decimal_place*)

Convert a float into the output string with a specific number of decimal place

Parameters

| | | |
|-----------|----------------------|--|
| <i>in</i> | <i>output</i> | the output sting |
| <i>in</i> | <i>input</i> | the input float |
| <i>in</i> | <i>decimal_place</i> | the number of decimal, must be between 0 and 3 |

Returns

the ranking

4.23.3.5 bool convertStringBool (char * *str*)

Convert a string into a bool

Parameters

| | | |
|-----------|------------|------------|
| <i>in</i> | <i>str</i> | the string |
|-----------|------------|------------|

Returns

the bool

4.23.3.6 float convertStringFloat (char * *str*)

Convert a string into a float

Parameters

| | | |
|-----------|------------|------------|
| <i>in</i> | <i>str</i> | the string |
|-----------|------------|------------|

Returns

the float

4.23.3.7 int convertStringInt (char * *str*)

Convert a string into a int

Parameters

| | | |
|-----------|------------|------------|
| <i>in</i> | <i>str</i> | the string |
|-----------|------------|------------|

Returns

the int

4.23.3.8 char* integerToYesNo (int *i*, char * *yes*, char * *no*)

Transform an integer to yes or no

Parameters

| | | |
|-----------|------------|----------------|
| <i>in</i> | <i>i</i> | the integer |
| <i>in</i> | <i>yes</i> | the yes string |
| <i>in</i> | <i>no</i> | the no string |

Returns

yes if $i > 0$, no otherwise

4.23.3.9 void libcsuper_initialize ()

Initialize libcsuper with gettext.

4.23.3.10 void* myAlloc (int *size_alloue*)

Allocate a memory block and check if everything is OK.

Parameters

| | | |
|-----------|--------------------|----------|
| <i>in</i> | <i>size_alloue</i> | the size |
|-----------|--------------------|----------|

Returns

a pointer on the allocate memory block

Here is the call graph for this function:



4.23.3.11 void myRealloc (void ** *ptr*, int *size_alloue*)

Here is the call graph for this function:



4.23.3.12 char* utf8ToLatin9 (const char *const *string*)

Create a dynamically allocated copy of string, changing the encoding from UTF-8 to ISO-8859-15. Unsupported code points are ignored.

Parameters

| | | |
|----|---------------|---------------------------|
| in | <i>string</i> | the input string in UTF-8 |
|----|---------------|---------------------------|

Returns

a newly allocated string in ISO-8859-15

4.23.3.13 void wrongChoice ()

Display an error message.

Here is the call graph for this function:



Index

- - share.h, [181](#)
- about
 - toolbar_button_preferences_struct, [17](#)
- addConfigListFile
 - game_config.c, [119](#)
 - game_config.h, [128](#)
- addConfigListGroupConfig
 - game_config.c, [120](#)
 - game_config.h, [128](#)
- addDistributorCsuStruct
 - csu_struct.c, [60](#)
 - csu_struct.h, [71](#)
- addFileCsuExtension
 - filename.c, [109](#)
 - filename.h, [114](#)
- addFileCsvExtension
 - filename.c, [110](#)
 - filename.h, [115](#)
- addFileDatExtension
 - filename.c, [110](#)
 - filename.h, [115](#)
- addFileExtension
 - filename.c, [110](#)
 - filename.h, [115](#)
- addFileGnuplotExtension
 - filename.c, [111](#)
 - filename.h, [116](#)
- addFilePdfExtension
 - filename.c, [111](#)
 - filename.h, [116](#)
- addPodiumPdf
 - export.c, [82](#)
 - export.h, [93](#)
- addStatsPdf
 - export.c, [82](#)
 - export.h, [93](#)
- addTotalPointsRankingPdf
 - export.c, [83](#)
 - export.h, [94](#)
- addXmlBoolNode
 - csu_files.c, [47](#)
 - csu_files.h, [54](#)
- addXmlFloatNode
 - csu_files.c, [47](#)
 - csu_files.h, [54](#)
- addXmlFloatNodeIntProp
 - csu_files.c, [47](#)
 - csu_files.h, [54](#)
- addXmlIntNode
 - csu_files.c, [48](#)
 - csu_files.h, [55](#)
- addXmlStringNode
 - csu_files.c, [48](#)
 - csu_files.h, [55](#)
- begin_score
 - game_config, [12](#)
- CONFIGURATION_FILE_NAME
 - game_config.h, [128](#)
- CONFIGURATION_FOLDER_NAME
 - game_config.h, [128](#)
- CONFIGURATION_XML_FILENAME
 - game_config.h, [128](#)
- CSUPER_VERSION
 - share.h, [181](#)
- calculateFromString
 - calculator.c, [22](#)
 - calculator.h, [34](#)
- calculator
 - main_window_side, [14](#)
- calculator.c, [21](#)
 - calculateFromString, [22](#)
 - calculatorListCalculate, [22](#)
 - calculatorListDelete, [23](#)
 - calculatorListDeleteCurrent, [23](#)
 - calculatorListEmpty, [25](#)
 - calculatorListFirst, [25](#)
 - calculatorListGetCurrentNumber, [27](#)
 - calculatorListGetCurrentOperator, [27](#)
 - calculatorListInit, [27](#)
 - calculatorListInsertAfterCurrent, [27](#)
 - calculatorListInsertBeforeCurrent, [28](#)
 - calculatorListInsertLast, [29](#)
 - calculatorListInsertLastFromString, [29](#)
 - calculatorListIsCurrentNumber, [30](#)
 - calculatorListLast, [30](#)
 - calculatorListNext, [30](#)
 - calculatorListOne, [31](#)
 - calculatorListOutOfList, [31](#)
 - calculatorListPrevious, [31](#)
 - calculatorListPrint, [31](#)
 - calculatorListSetOnFirst, [32](#)
 - calculatorListSetOnLast, [32](#)
 - calculatorSearchNextOperatorString, [32](#)
 - deleteCalculatorNodeList, [32](#)
 - newCalculatorNodeList, [33](#)
- calculator.h, [33](#)

- calculateFromString, 34
- calculatorListCalculate, 35
- calculatorListDelete, 36
- calculatorListDeleteCurrent, 36
- calculatorListEmpty, 38
- calculatorListGetCurrentNumber, 38
- calculatorListGetCurrentOperator, 40
- calculatorListInit, 40
- calculatorListInsertAfterCurrent, 40
- calculatorListInsertBeforeCurrent, 41
- calculatorListInsertLast, 42
- calculatorListInsertLastFromString, 42
- calculatorListIsCurrentNumber, 43
- calculatorListLast, 43
- calculatorListLirst, 43
- calculatorListNext, 43
- calculatorListOne, 44
- calculatorListOutOfList, 44
- calculatorListPrevious, 44
- calculatorListPrint, 44
- calculatorListSetOnFirst, 45
- calculatorListSetOnLast, 45
- calculatorNodeList, 34
- calculatorSearchNextOperatorString, 45
- deleteCalculatorNodeList, 45
- newCalculatorNodeList, 46
- calculatorList, 5
 - current, 5
 - first, 5
 - last, 5
- calculatorListCalculate
 - calculator.c, 22
 - calculator.h, 35
- calculatorListDelete
 - calculator.c, 23
 - calculator.h, 36
- calculatorListDeleteCurrent
 - calculator.c, 23
 - calculator.h, 36
- calculatorListEmpty
 - calculator.c, 25
 - calculator.h, 38
- calculatorListFirst
 - calculator.c, 25
- calculatorListGetCurrentNumber
 - calculator.c, 27
 - calculator.h, 38
- calculatorListGetCurrentOperator
 - calculator.c, 27
 - calculator.h, 40
- calculatorListInit
 - calculator.c, 27
 - calculator.h, 40
- calculatorListInsertAfterCurrent
 - calculator.c, 27
 - calculator.h, 40
- calculatorListInsertBeforeCurrent
 - calculator.c, 28
- calculator.h, 41
- calculatorListInsertLast
 - calculator.c, 29
 - calculator.h, 42
- calculatorListInsertLastFromString
 - calculator.c, 29
 - calculator.h, 42
- calculatorListIsCurrentNumber
 - calculator.c, 30
 - calculator.h, 43
- calculatorListLast
 - calculator.c, 30
 - calculator.h, 43
- calculatorListLirst
 - calculator.h, 43
- calculatorListNext
 - calculator.c, 30
 - calculator.h, 43
- calculatorListOne
 - calculator.c, 31
 - calculator.h, 44
- calculatorListOutOfList
 - calculator.c, 31
 - calculator.h, 44
- calculatorListPrevious
 - calculator.c, 31
 - calculator.h, 44
- calculatorListPrint
 - calculator.c, 31
 - calculator.h, 44
- calculatorListSetOnFirst
 - calculator.c, 32
 - calculator.h, 45
- calculatorListSetOnLast
 - calculator.c, 32
 - calculator.h, 45
- calculatorNodeList, 5
 - calculator.h, 34
 - is_number, 6
 - next, 6
 - number, 6
 - operator, 6
 - previous, 6
- calculatorSearchNextOperatorString
 - calculator.c, 32
 - calculator.h, 45
- canUseUtf8Pdf
 - export.c, 83
 - export.h, 94
- changeDistributor
 - csu_struct.c, 61
 - csu_struct.h, 72
- changeSystemPath
 - preferences_files.c, 148
 - preferences_files.h, 163
- CharacterSetPdf
 - export.h, 93
- charset

- export_pdf_preferences, 11
- chart
 - one_preferences, 15
 - preferences, 16
- chart_exportation, 6
 - height, 7
 - total_points, 7
 - width, 7
- chart_type
 - preferences_files.h, 163
- checkFilename
 - filename.c, 111
 - filename.h, 116
- checkPath
 - filename.c, 112
 - filename.h, 117
- clearScreen
 - share.c, 176
 - share.h, 181
- closeCsuStruct
 - csu_struct.c, 61
 - csu_struct.h, 72
- closeExportPdf
 - export.c, 84
 - export.h, 94
- closeFile
 - file.c, 104
 - file.h, 106
- closeListGroupConfig
 - game_config.c, 120
 - game_config.h, 129
- compareFloatAscending
 - share.c, 176
 - share.h, 181
- compareFloatDescending
 - share.c, 177
 - share.h, 181
- config
 - csuStruct, 7
- consecutive
 - difference_between_player, 9
- convertFloatString
 - share.c, 177
 - share.h, 182
- convertStringBool
 - share.c, 177
 - share.h, 182
- convertStringFloat
 - share.c, 177
 - share.h, 182
- convertStringInt
 - share.c, 178
 - share.h, 182
- copy
 - toolbar_button_preferences_struct, 17
- copyCsuStruct
 - csu_struct.c, 61
 - csu_struct.h, 72
- createFileChartExportation
 - preferences_files.c, 149
 - preferences_files.h, 164
- createFileDifferenceBetweenPlayer
 - preferences_files.c, 149
 - preferences_files.h, 164
- createFileMainWidowSize
 - preferences_files.c, 150
 - preferences_files.h, 165
- createFileMainWindowSide
 - preferences_files.c, 150
 - preferences_files.h, 165
- createFilePdfPreferences
 - preferences_files.c, 151
 - preferences_files.h, 166
- createFileScoreDisplay
 - preferences_files.c, 151
 - preferences_files.h, 166
- createFileSystemPath
 - preferences_files.c, 152
 - preferences_files.h, 167
- createFileToolBarButtonPreferences
 - preferences_files.c, 152
 - preferences_files.h, 167
- createFirstPagePdf
 - export.c, 84
 - export.h, 96
- createOtherPagePdf
 - export.c, 84
 - export.h, 96
- createPdfGrid
 - export.c, 85
 - export.h, 97
- createPreferencesFolder
 - preferences_files.c, 153
 - preferences_files.h, 168
- csu_files.c, 46
 - addXmlBoolNode, 47
 - addXmlFloatNode, 47
 - addXmlFloatNodeIntProp, 47
 - addXmlIntNode, 48
 - addXmlStringNode, 48
 - openFileCsuExtension, 48
 - readCsuFile, 49
 - readCsuXmlFile, 49
 - writeCsuFile, 50
 - writeCsuXmlFile, 50
 - writeFileNewTurn, 52
- csu_files.h, 52
 - addXmlBoolNode, 54
 - addXmlFloatNode, 54
 - addXmlFloatNodeIntProp, 54
 - addXmlIntNode, 55
 - addXmlStringNode, 55
 - FILE_EXTENSION_CSU, 53
 - openFileCsuExtension, 55
 - readCsuFile, 56
 - readCsuXmlFile, 56

- SIZE_MAX_FILE_NAME, 53
- STRING_CHECK_CSU_FILE, 53
- writeCsuFile, 57
- writeCsuXmlFile, 57
- writeFileNewTurn, 59
- csu_struct.c, 59
 - addDistributorCsuStruct, 60
 - changeDistributor, 61
 - closeCsuStruct, 61
 - copyCsuStruct, 61
 - deleteTurn, 62
 - diffrentsPlayerName, 62
 - endNewTurn, 62
 - exceedMaxNumber, 64
 - lastRankAtTurn, 64
 - maxNbTurn, 65
 - meanPoints, 65
 - nbTurnBest, 65
 - nbTurnFirst, 66
 - nbTurnLast, 66
 - nbTurnWorst, 67
 - newCsuStruct, 67
 - pointsAtTurn, 67
 - rankAtTurn, 68
 - rankCalculation, 68
 - searchIndexFromPosition, 69
 - searchPlayerIndex, 69
 - startNewTurn, 70
- csu_struct.h, 70
 - addDistributorCsuStruct, 71
 - changeDistributor, 72
 - closeCsuStruct, 72
 - copyCsuStruct, 72
 - deleteTurn, 73
 - diffrentsPlayerName, 73
 - endNewTurn, 73
 - exceedMaxNumber, 75
 - lastRankAtTurn, 75
 - maxNbTurn, 76
 - meanPoints, 76
 - nbTurnBest, 76
 - nbTurnFirst, 77
 - nbTurnLast, 77
 - nbTurnWorst, 78
 - newCsuStruct, 78
 - pointsAtTurn, 78
 - rankAtTurn, 79
 - rankCalculation, 79
 - SIZE_MAX_NAME, 71
 - searchIndexFromPosition, 80
 - searchPlayerIndex, 80
 - startNewTurn, 81
 - VERSION, 71
- csuStruct, 7
 - config, 7
 - day, 7
 - distributor, 8
 - month, 8
 - nb_player, 8
 - nb_turn, 8
 - player_names, 8
 - point, 8
 - rank, 8
 - size_max_name, 8
 - total_points, 8
 - version, 8
 - year, 8
- current
 - calculatorList, 5
- cut
 - toolbar_button_preferences_struct, 17
- DEFAULT_FONT_SIZE
 - export.h, 93
- DEFAULT_MARGIN
 - export.h, 93
- day
 - csuStruct, 7
- decimal_place
 - game_config, 12
- delete
 - toolbar_button_preferences_struct, 17
- delete_file
 - toolbar_button_preferences_struct, 17
- deleteCalculatorNodeList
 - calculator.c, 32
 - calculator.h, 45
- deleteFile
 - file.c, 104
 - file.h, 107
- deleteTurn
 - csu_struct.c, 62
 - csu_struct.h, 73
- diff
 - one_preferences, 15
 - preferences, 16
- diff_type
 - preferences_files.h, 163
- difference_between_player, 9
 - consecutive, 9
 - first, 9
 - last, 9
- diffrentsChartExportationStruct
 - preferences_files.c, 153
 - preferences_files.h, 168
- diffrentsPlayerName
 - csu_struct.c, 62
 - csu_struct.h, 73
- diffrentsTExportPdfPreferencesStruct
 - preferences_files.c, 153
 - preferences_files.h, 168
- diffrentsToolBarButtonPreferencesStruct
 - preferences_files.c, 154
 - preferences_files.h, 169
- direction
 - export_pdf_preferences, 11
- displayHelp

- main_argument.c, [143](#)
- main_argument.h, [147](#)
- distributor
 - csuStruct, [8](#)
- edit_suppr
 - score_display, [16](#)
- endNewTurn
 - csu_struct.c, [62](#)
 - csu_struct.h, [73](#)
- errorHandler
 - export.c, [85](#)
 - export.h, [97](#)
- exceedMaxNumber
 - csu_struct.c, [64](#)
 - csu_struct.h, [75](#)
- export
 - toolbar_button_preferences_struct, [18](#)
- export.c, [81](#)
 - addPodiumPdf, [82](#)
 - addStatsPdf, [82](#)
 - addTotalPointsRankingPdf, [83](#)
 - canUseUtf8Pdf, [83](#)
 - closeExportPdf, [84](#)
 - createFirstPagePdf, [84](#)
 - createOtherPagePdf, [84](#)
 - createPdfGrid, [85](#)
 - errorHandler, [85](#)
 - exportToCsv, [86](#)
 - exportToM, [86](#)
 - exportToPdf, [87](#)
 - initializePdfExport, [88](#)
 - pdfShowText, [88](#)
 - pdfTextOutTable, [89](#)
 - printLegendPdf, [89](#)
 - printNamesPdf, [90](#)
 - printPointsPdf, [90](#)
 - tableWidthCalculatePdf, [91](#)
- export.h, [91](#)
 - addPodiumPdf, [93](#)
 - addStatsPdf, [93](#)
 - addTotalPointsRankingPdf, [94](#)
 - canUseUtf8Pdf, [94](#)
 - CharacterSetPdf, [93](#)
 - closeExportPdf, [94](#)
 - createFirstPagePdf, [96](#)
 - createOtherPagePdf, [96](#)
 - createPdfGrid, [97](#)
 - DEFAULT_FONT_SIZE, [93](#)
 - DEFAULT_MARGIN, [93](#)
 - errorHandler, [97](#)
 - exportToCsv, [97](#)
 - exportToM, [98](#)
 - exportToPdf, [99](#)
 - ISO885915, [93](#)
 - initializePdfExport, [100](#)
 - pdfShowText, [100](#)
 - pdfTextOutTable, [101](#)
 - printLegendPdf, [101](#)
 - printNamesPdf, [102](#)
 - printPointsPdf, [102](#)
 - TEXT_BUFFER_SIZE, [93](#)
 - tableWidthCalculatePdf, [103](#)
 - UTF8, [93](#)
- export_pdf, [9](#)
 - font, [10](#)
 - line, [10](#)
 - line_height, [10](#)
 - num_page, [10](#)
 - pdf, [10](#)
 - pref, [10](#)
 - stat_print, [10](#)
 - table_line_height, [10](#)
 - total_points_ranking_print, [10](#)
- export_pdf_preferences, [10](#)
 - charset, [11](#)
 - direction, [11](#)
 - font_size, [11](#)
 - margin, [11](#)
 - pdf_size_for_chart, [11](#)
 - ranking_turn, [11](#)
 - size, [11](#)
 - total_points_turn, [11](#)
- export_to_csv
 - main_argument.h, [147](#)
- export_to_gnuplot
 - main_argument.h, [146](#)
- export_to_m
 - main_argument.h, [147](#)
- export_to_pdf
 - main_argument.h, [146](#)
- exportConfigFile
 - game_config.c, [120](#)
 - game_config.h, [129](#)
- exportToCsv
 - export.c, [86](#)
 - export.h, [97](#)
- exportToGnuplotData
 - gnuplot.c, [138](#)
 - gnuplot.h, [140](#)
- exportToGnuplotFile
 - gnuplot.c, [138](#)
 - gnuplot.h, [141](#)
- exportToGnuplotScript
 - gnuplot.c, [139](#)
 - gnuplot.h, [142](#)
- exportToM
 - export.c, [86](#)
 - export.h, [98](#)
- exportToPdf
 - export.c, [87](#)
 - export.h, [99](#)
- exportation_preferences
 - toolbar_button_preferences_struct, [18](#)
- FILE_EXTENSION_CSU
 - csu_files.h, [53](#)
- FILENAME_DIFFERENCE_BETWEEN_PLAYER

- preferences_files.h, [162](#)
- FILENAME_MAIN_WINDOW_SIDE
 - preferences_files.h, [162](#)
- FILENAME_MAIN_WINDOW_SIZE
 - preferences_files.h, [162](#)
- FILENAME_PDF_PREFERENCES
 - preferences_files.h, [162](#)
- FILENAME_PREFERENCES_XML
 - preferences_files.h, [162](#)
- FILENAME_SCORE_DISPLAY
 - preferences_files.h, [162](#)
- FILENAME_SYSTEM_PATH
 - preferences_files.h, [163](#)
- FILENAME_TOOLBAR_BUTTON_PREFERENCES
 - preferences_files.h, [163](#)
- file.c, [103](#)
 - closeFile, [104](#)
 - deleteFile, [104](#)
 - openFile, [105](#)
 - readFileSize, [105](#)
 - renameFile, [105](#)
- file.h, [106](#)
 - closeFile, [106](#)
 - deleteFile, [107](#)
 - openFile, [107](#)
 - readFileSize, [108](#)
 - renameFile, [108](#)
- filename.c, [109](#)
 - addFileCsuExtension, [109](#)
 - addFileCsvExtension, [110](#)
 - addFileDatExtension, [110](#)
 - addFileExtension, [110](#)
 - addFileGnuplotExtension, [111](#)
 - addFilePdfExtension, [111](#)
 - checkFilename, [111](#)
 - checkPath, [112](#)
 - getFolderFromFilename, [112](#)
 - getSimpleFilenameFromFullFilename, [113](#)
 - readHomePath, [113](#)
 - readHomePathSlash, [113](#)
 - removeFileExtension, [113](#)
 - removeFilenameExtension, [113](#)
- filename.h, [114](#)
 - addFileCsuExtension, [114](#)
 - addFileCsvExtension, [115](#)
 - addFileDatExtension, [115](#)
 - addFileExtension, [115](#)
 - addFileGnuplotExtension, [116](#)
 - addFilePdfExtension, [116](#)
 - checkFilename, [116](#)
 - checkPath, [117](#)
 - getFolderFromFilename, [117](#)
 - getSimpleFilenameFromFullFilename, [118](#)
 - readHomePath, [118](#)
 - readHomePathSlash, [118](#)
 - removeFileExtension, [118](#)
 - removeFilenameExtension, [118](#)
- first
 - calculatorList, [5](#)
 - difference_between_player, [9](#)
- first_way
 - game_config, [12](#)
- font
 - export_pdf, [10](#)
- font_size
 - export_pdf_preferences, [11](#)
- GAME_CONFIG_FILE_XML_VERSION
 - game_config.h, [128](#)
- game_config, [12](#)
 - begin_score, [12](#)
 - decimal_place, [12](#)
 - first_way, [12](#)
 - max, [12](#)
 - name, [12](#)
 - nb_max, [12](#)
 - turn_based, [12](#)
 - use_distributor, [12](#)
- game_config.c, [119](#)
 - addConfigListFile, [119](#)
 - addConfigListGameConfig, [120](#)
 - closeListGameConfig, [120](#)
 - exportConfigFile, [120](#)
 - importConfigFile, [121](#)
 - makeConfigListFile, [121](#)
 - newConfigFile, [122](#)
 - newListGameConfig, [122](#)
 - newListGameConfigFromImport, [123](#)
 - readConfigFile, [123](#)
 - readConfigListFile, [124](#)
 - readXmlListGameConfig, [124](#)
 - readXmlListGameConfigWithId, [124](#)
 - removeConfigFile, [125](#)
 - removeConfigListFile, [125](#)
 - removeConfigListGameConfig, [125](#)
 - writeXmlListGameConfig, [126](#)
 - writeXmlListGameConfigWithId, [126](#)
- game_config.h, [126](#)
 - addConfigListFile, [128](#)
 - addConfigListGameConfig, [128](#)
 - CONFIGURATION_FILE_NAME, [128](#)
 - CONFIGURATION_FOLDER_NAME, [128](#)
 - CONFIGURATION_XML_FILENAME, [128](#)
 - closeListGameConfig, [129](#)
 - exportConfigFile, [129](#)
 - GAME_CONFIG_FILE_XML_VERSION, [128](#)
 - importConfigFile, [129](#)
 - makeConfigListFile, [130](#)
 - newConfigFile, [130](#)
 - newListGameConfig, [132](#)
 - newListGameConfigFromImport, [132](#)
 - readConfigFile, [133](#)
 - readConfigListFile, [133](#)
 - readXmlListGameConfig, [134](#)
 - readXmlListGameConfigWithId, [134](#)
 - removeConfigFile, [134](#)
 - removeConfigListFile, [136](#)

- removeConfigListGameConfig, 136
 - STRING_CHECK_GAME_CONFIG, 128
 - writeXmlListGameConfig, 136
 - writeXmlListGameConfigWithId, 137
- game_configs
 - list_game_config, 13
- game_configuration_preferences
 - toolbar_button_preferences_struct, 18
- game_information
 - main_window_side, 14
- getFolderFromFilename
 - filename.c, 112
 - filename.h, 117
- getSimpleFilenameFromFullFilename
 - filename.c, 113
 - filename.h, 118
- gnuplot.c, 137
 - exportToGnuplotData, 138
 - exportToGnuplotFile, 138
 - exportToGnuplotScript, 139
- gnuplot.h, 140
 - exportToGnuplotData, 140
 - exportToGnuplotFile, 141
 - exportToGnuplotScript, 142
- height
 - chart_exportation, 7
 - main_window_size, 14
- help
 - main_argument.h, 146
- ISO885915
 - export.h, 93
- importConfigFile
 - game_config.c, 121
 - game_config.h, 129
- initializePdfExport
 - export.c, 88
 - export.h, 100
- integerToYesNo
 - share.c, 178
 - share.h, 183
- is_maximize
 - main_window_size, 14
- is_number
 - calculatorNodeList, 6
- last
 - calculatorList, 5
 - difference_between_player, 9
- lastRankAtTurn
 - csu_struct.c, 64
 - csu_struct.h, 75
- libcsuper.h, 142
 - NOT_LIBCSUPER, 143
- libcsuper_initialize
 - share.c, 178
 - share.h, 183
- line
 - export_pdf, 10
- line_height
 - export_pdf, 10
- list_game_config, 13
 - game_configs, 13
 - name_game_config, 13
 - nb_config, 13
- main_argument.c, 143
 - displayHelp, 143
 - searchArgument, 144
- main_argument.h, 144
 - displayHelp, 147
 - export_to_csv, 147
 - export_to_gnuplot, 146
 - export_to_m, 147
 - export_to_pdf, 146
 - help, 146
 - main_argument_function, 146
 - open_file, 146
 - read_file, 146
 - STRING_EXPORT_TO_CSV, 145
 - STRING_EXPORT_TO_CSV_RED, 145
 - STRING_EXPORT_TO_GNUPLOT, 145
 - STRING_EXPORT_TO_GNUPLOT_RED, 145
 - STRING_EXPORT_TO_M, 145
 - STRING_EXPORT_TO_M_RED, 146
 - STRING_EXPORT_TO_PDF, 146
 - STRING_EXPORT_TO_PDF_RED, 146
 - STRING_HELP, 146
 - STRING_HELP_RED, 146
 - STRING_OPEN_FILE, 146
 - STRING_OPEN_FILE_RED, 146
 - STRING_READ_FILE, 146
 - STRING_READ_FILE_RED, 146
 - searchArgument, 147
- main_argument_function
 - main_argument.h, 146
- main_window_side, 13
 - calculator, 14
 - game_information, 14
 - ranking, 14
- main_window_size, 14
 - height, 14
 - is_maximize, 14
 - width, 14
- makeConfigListFile
 - game_config.c, 121
 - game_config.h, 130
- margin
 - export_pdf_preferences, 11
- max
 - game_config, 12
- maxNbTurn
 - csu_struct.c, 65
 - csu_struct.h, 76
- meanPoints
 - csu_struct.c, 65
 - csu_struct.h, 76

- month
 - csuStruct, 8
- myAlloc
 - share.c, 178
 - share.h, 183
- myRealloc
 - share.c, 179
 - share.h, 183
- NOT_LIBCSUPER
 - libcsuper.h, 143
- name
 - game_config, 12
- name_game_config
 - list_game_config, 13
- nb_config
 - list_game_config, 13
- nb_max
 - game_config, 12
- nb_player
 - csuStruct, 8
- nb_turn
 - csuStruct, 8
- nbTurnBest
 - csu_struct.c, 65
 - csu_struct.h, 76
- nbTurnFirst
 - csu_struct.c, 66
 - csu_struct.h, 77
- nbTurnLast
 - csu_struct.c, 66
 - csu_struct.h, 77
- nbTurnWorst
 - csu_struct.c, 67
 - csu_struct.h, 78
- new
 - toolbar_button_preferences_struct, 18
- newCalculatorNodeList
 - calculator.c, 33
 - calculator.h, 46
- newConfigFile
 - game_config.c, 122
 - game_config.h, 130
- newCsuStruct
 - csu_struct.c, 67
 - csu_struct.h, 78
- newListGameConfig
 - game_config.c, 122
 - game_config.h, 132
- newListGameConfigFromImport
 - game_config.c, 123
 - game_config.h, 132
- next
 - calculatorNodeList, 6
- num_page
 - export_pdf, 10
- number
 - calculatorNodeList, 6
- one_preferences, 15
 - chart, 15
 - diff, 15
 - pdf, 15
 - score, 15
 - side, 15
 - size, 15
 - toolbar, 15
- open
 - toolbar_button_preferences_struct, 18
- open_file
 - main_argument.h, 146
- openFile
 - file.c, 105
 - file.h, 107
- openFileCsuExtension
 - csu_files.c, 48
 - csu_files.h, 55
- operator
 - calculatorNodeList, 6
- PREFERENCES_FILE_XML_VERSION
 - preferences_files.h, 163
- PREFERENCES_FOLDER_NAME
 - preferences_files.h, 163
- paste
 - toolbar_button_preferences_struct, 18
- pdf
 - export_pdf, 10
 - one_preferences, 15
 - preferences, 16
- pdf_size_for_chart
 - export_pdf_preferences, 11
- pdf_type
 - preferences_files.h, 163
- pdfShowText
 - export.c, 88
 - export.h, 100
- pdfTextOutTable
 - export.c, 89
 - export.h, 101
- player_names
 - csuStruct, 8
- point
 - csuStruct, 8
- pointsAtTurn
 - csu_struct.c, 67
 - csu_struct.h, 78
- pref
 - export_pdf, 10
- preferences, 15
 - chart, 16
 - diff, 16
 - pdf, 16
 - score, 16
 - side, 16
 - size, 16
 - toolbar, 16
 - toolbar_button_preferences_struct, 18

- preferences_files.c, 147
 - changeSystemPath, 148
 - createFileChartExportation, 149
 - createFileDifferenceBetweenPlayer, 149
 - createFileMainWidowSize, 150
 - createFileMainWindowSide, 150
 - createFilePdfPreferences, 151
 - createFileScoreDisplay, 151
 - createFileSystemPath, 152
 - createFileToolBarButtonPreferences, 152
 - createPreferencesFolder, 153
 - differeentsChartExportationStruct, 153
 - differeentsTExportPdfPreferencesStruct, 153
 - differeentsToolBarButtonPreferencesStruct, 154
 - readFileChartExportation, 154
 - readFileDifferenceBetweenPlayer, 154
 - readFileMainWidowSize, 155
 - readFileMainWindowSide, 155
 - readFilePdfPreferences, 156
 - readFileScoreDisplay, 156
 - readFileSystemPath, 157
 - readFileToolBarButtonPreferences, 157
 - readSystemPath, 158
 - readXmlPreferencesFile, 158
 - readXmlPreferencesFileType, 159
 - writeXmlPreferencesFile, 159
 - writeXmlPreferencesFileType, 160
- preferences_files.h, 161
 - changeSystemPath, 163
 - chart_type, 163
 - createFileChartExportation, 164
 - createFileDifferenceBetweenPlayer, 164
 - createFileMainWidowSize, 165
 - createFileMainWindowSide, 165
 - createFilePdfPreferences, 166
 - createFileScoreDisplay, 166
 - createFileSystemPath, 167
 - createFileToolBarButtonPreferences, 167
 - createPreferencesFolder, 168
 - diff_type, 163
 - differeentsChartExportationStruct, 168
 - differeentsTExportPdfPreferencesStruct, 168
 - differeentsToolBarButtonPreferencesStruct, 169
 - FILENAME_DIFFERENCE_BETWEEN_PLAYER, 162
 - FILENAME_MAIN_WINDOW_SIDE, 162
 - FILENAME_MAIN_WINDOW_SIZE, 162
 - FILENAME_PDF_PREFERENCES, 162
 - FILENAME_PREFERENCES_XML, 162
 - FILENAME_SCORE_DISPLAY, 162
 - FILENAME_SYSTEM_PATH, 163
 - FILENAME_TOOLBAR_BUTTON_PREFERENCES, 163
 - PREFERENCES_FILE_XML_VERSION, 163
 - PREFERENCES_FOLDER_NAME, 163
 - pdf_type, 163
 - preferences_type, 163
 - readFileChartExportation, 169
 - readFileDifferenceBetweenPlayer, 169
 - readFileMainWidowSize, 170
 - readFileMainWindowSide, 170
 - readFilePdfPreferences, 171
 - readFileScoreDisplay, 171
 - readFileSystemPath, 172
 - readFileToolBarButtonPreferences, 172
 - readSystemPath, 173
 - readXmlPreferencesFile, 173
 - readXmlPreferencesFileType, 174
 - score_type, 163
 - side_type, 163
 - size_type, 163
 - toolbar_type, 163
 - writeXmlPreferencesFile, 174
 - writeXmlPreferencesFileType, 175
- preferences_type
 - preferences_files.h, 163
- previous
 - calculatorNodeList, 6
- printLegendPdf
 - export.c, 89
 - export.h, 101
- printNamesPdf
 - export.c, 90
 - export.h, 102
- printPointsPdf
 - export.c, 90
 - export.h, 102
- properties
 - toolbar_button_preferences_struct, 18
- rank
 - csuStruct, 8
- rankAtTurn
 - csu_struct.c, 68
 - csu_struct.h, 79
- rankCalculation
 - csu_struct.c, 68
 - csu_struct.h, 79
- ranking
 - main_window_side, 14
 - score_display, 16
- ranking_turn
 - export_pdf_preferences, 11
- read_file
 - main_argument.h, 146
- readConfigFile
 - game_config.c, 123
 - game_config.h, 133
- readConfigListFile
 - game_config.c, 124
 - game_config.h, 133
- readCsuFile
 - csu_files.c, 49
 - csu_files.h, 56
- readCsuXmlFile
 - csu_files.c, 49
 - csu_files.h, 56

- readFileChartExportation
 - preferences_files.c, [154](#)
 - preferences_files.h, [169](#)
- readFileDifferenceBetweenPlayer
 - preferences_files.c, [154](#)
 - preferences_files.h, [169](#)
- readFileMainWidowSize
 - preferences_files.c, [155](#)
 - preferences_files.h, [170](#)
- readFileMainWindowSide
 - preferences_files.c, [155](#)
 - preferences_files.h, [170](#)
- readFilePdfPreferences
 - preferences_files.c, [156](#)
 - preferences_files.h, [171](#)
- readFileScoreDisplay
 - preferences_files.c, [156](#)
 - preferences_files.h, [171](#)
- readFileSize
 - file.c, [105](#)
 - file.h, [108](#)
- readFileSystemPath
 - preferences_files.c, [157](#)
 - preferences_files.h, [172](#)
- readFileToolBarButtonPreferences
 - preferences_files.c, [157](#)
 - preferences_files.h, [172](#)
- readHomePath
 - filename.c, [113](#)
 - filename.h, [118](#)
- readHomePathSlash
 - filename.c, [113](#)
 - filename.h, [118](#)
- readSystemPath
 - preferences_files.c, [158](#)
 - preferences_files.h, [173](#)
- readXmlListGroupConfig
 - game_config.c, [124](#)
 - game_config.h, [134](#)
- readXmlListGroupConfigWithId
 - game_config.c, [124](#)
 - game_config.h, [134](#)
- readXmlPreferencesFile
 - preferences_files.c, [158](#)
 - preferences_files.h, [173](#)
- readXmlPreferencesFileType
 - preferences_files.c, [159](#)
 - preferences_files.h, [174](#)
- redo
 - toolbar_button_preferences_struct, [18](#)
- removeConfigFile
 - game_config.c, [125](#)
 - game_config.h, [134](#)
- removeConfigListFile
 - game_config.c, [125](#)
 - game_config.h, [136](#)
- removeConfigListGroupConfig
 - game_config.c, [125](#)
- game_config.h, [136](#)
- removeFileExtension
 - filename.c, [113](#)
 - filename.h, [118](#)
- removeFilenameExtension
 - filename.c, [113](#)
 - filename.h, [118](#)
- renameFile
 - file.c, [105](#)
 - file.h, [108](#)
- SIZE_MAX_FILE_NAME
 - csu_files.h, [53](#)
- SIZE_MAX_NAME
 - csu_struct.h, [71](#)
- STRING_CHECK_CSU_FILE
 - csu_files.h, [53](#)
- STRING_CHECK_GAME_CONFIG
 - game_config.h, [128](#)
- STRING_EXPORT_TO_CSV
 - main_argument.h, [145](#)
- STRING_EXPORT_TO_CSV_RED
 - main_argument.h, [145](#)
- STRING_EXPORT_TO_GNUPLOT
 - main_argument.h, [145](#)
- STRING_EXPORT_TO_GNUPLOT_RED
 - main_argument.h, [145](#)
- STRING_EXPORT_TO_M
 - main_argument.h, [145](#)
- STRING_EXPORT_TO_M_RED
 - main_argument.h, [146](#)
- STRING_EXPORT_TO_PDF
 - main_argument.h, [146](#)
- STRING_EXPORT_TO_PDF_RED
 - main_argument.h, [146](#)
- STRING_HELP
 - main_argument.h, [146](#)
- STRING_HELP_RED
 - main_argument.h, [146](#)
- STRING_OPEN_FILE
 - main_argument.h, [146](#)
- STRING_OPEN_FILE_RED
 - main_argument.h, [146](#)
- STRING_READ_FILE
 - main_argument.h, [146](#)
- STRING_READ_FILE_RED
 - main_argument.h, [146](#)
- save_as
 - toolbar_button_preferences_struct, [18](#)
- score
 - one_preferences, [15](#)
 - preferences, [16](#)
- score_display, [16](#)
 - edit_suppr, [16](#)
 - ranking, [16](#)
 - total_points, [16](#)
- score_type
 - preferences_files.h, [163](#)
- searchArgument

- main_argument.c, [144](#)
- main_argument.h, [147](#)
- searchIndexFromPosition
 - csu_struct.c, [69](#)
 - csu_struct.h, [80](#)
- searchPlayerIndex
 - csu_struct.c, [69](#)
 - csu_struct.h, [80](#)
- separator_1
 - toolbar_button_preferences_struct, [18](#)
- separator_2
 - toolbar_button_preferences_struct, [18](#)
- separator_3
 - toolbar_button_preferences_struct, [19](#)
- separator_4
 - toolbar_button_preferences_struct, [19](#)
- separator_5
 - toolbar_button_preferences_struct, [19](#)
- separator_6
 - toolbar_button_preferences_struct, [19](#)
- share.c, [176](#)
 - clearScreen, [176](#)
 - compareFloatAscending, [176](#)
 - compareFloatDescending, [177](#)
 - convertFloatString, [177](#)
 - convertStringBool, [177](#)
 - convertStringFloat, [177](#)
 - convertStringInt, [178](#)
 - integerToYesNo, [178](#)
 - libcsuper_initialize, [178](#)
 - myAlloc, [178](#)
 - myRealloc, [179](#)
 - utf8ToLatin9, [179](#)
 - wrongChoice, [179](#)
- share.h, [180](#)
 - _, [181](#)
 - CSUPER_VERSION, [181](#)
 - clearScreen, [181](#)
 - compareFloatAscending, [181](#)
 - compareFloatDescending, [181](#)
 - convertFloatString, [182](#)
 - convertStringBool, [182](#)
 - convertStringFloat, [182](#)
 - convertStringInt, [182](#)
 - integerToYesNo, [183](#)
 - libcsuper_initialize, [183](#)
 - myAlloc, [183](#)
 - myRealloc, [183](#)
 - utf8ToLatin9, [184](#)
 - wrongChoice, [184](#)
- side
 - one_preferences, [15](#)
 - preferences, [16](#)
- side_type
 - preferences_files.h, [163](#)
- size
 - export_pdf_preferences, [11](#)
 - one_preferences, [15](#)
 - preferences, [16](#)
 - size_max_name
 - csuStruct, [8](#)
 - size_type
 - preferences_files.h, [163](#)
 - startNewTurn
 - csu_struct.c, [70](#)
 - csu_struct.h, [81](#)
 - stat_print
 - export_pdf, [10](#)
- TEXT_BUFFER_SIZE
 - export.h, [93](#)
- table_line_height
 - export_pdf, [10](#)
- tableWidthCalculatePdf
 - export.c, [91](#)
 - export.h, [103](#)
- toolbar
 - one_preferences, [15](#)
 - preferences, [16](#)
- toolbar_button_preferences
 - toolbar_button_preferences_struct, [19](#)
- toolbar_button_preferences_struct, [17](#)
 - about, [17](#)
 - copy, [17](#)
 - cut, [17](#)
 - delete, [17](#)
 - delete_file, [17](#)
 - export, [18](#)
 - exportation_preferences, [18](#)
 - game_configuration_preferences, [18](#)
 - new, [18](#)
 - open, [18](#)
 - paste, [18](#)
 - preferences, [18](#)
 - properties, [18](#)
 - redo, [18](#)
 - save_as, [18](#)
 - separator_1, [18](#)
 - separator_2, [18](#)
 - separator_3, [19](#)
 - separator_4, [19](#)
 - separator_5, [19](#)
 - separator_6, [19](#)
 - toolbar_button_preferences, [19](#)
 - undo, [19](#)
- toolbar_type
 - preferences_files.h, [163](#)
- total_points
 - chart_exportation, [7](#)
 - csuStruct, [8](#)
 - score_display, [16](#)
- total_points_ranking_print
 - export_pdf, [10](#)
- total_points_turn
 - export_pdf_preferences, [11](#)
- turn_based
 - game_config, [12](#)

UTF8

- export.h, [93](#)

undo

- toolbar_button_preferences_struct, [19](#)

use_distributor

- game_config, [12](#)

utf8ToLatin9

- share.c, [179](#)

- share.h, [184](#)

VERSION

- csu_struct.h, [71](#)

version

- csuStruct, [8](#)

width

- chart_exportation, [7](#)

- main_window_size, [14](#)

writeCsuFile

- csu_files.c, [50](#)

- csu_files.h, [57](#)

writeCsuXmlFile

- csu_files.c, [50](#)

- csu_files.h, [57](#)

writeFileNewTurn

- csu_files.c, [52](#)

- csu_files.h, [59](#)

writeXmlListGameConfig

- game_config.c, [126](#)

- game_config.h, [136](#)

writeXmlListGameConfigWithId

- game_config.c, [126](#)

- game_config.h, [137](#)

writeXmlPreferencesFile

- preferences_files.c, [159](#)

- preferences_files.h, [174](#)

writeXmlPreferencesFileType

- preferences_files.c, [160](#)

- preferences_files.h, [175](#)

wrongChoice

- share.c, [179](#)

- share.h, [184](#)

year

- csuStruct, [8](#)